VisualAge Pacbase

Middleware
Guide d'utilisation

Version 3.5

Note

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section "Notices" après la Table des Matières.

En application de votre contrat de licence, vous pouvez consulter ou télécharger la documentation de VisualAge Pacbase, régulièrement mise à jour, à partir de :

http://www.ibm.com/support/docview.wss?rs=37&context=SSEP67&uid=swg27005478

La section Catalogue dans la page d'accueil de la Documentation vous permet d'identifier la dernière édition disponible du présent document.

Deuxième édition (Juillet 2007)

La présente édition s'applique à :

- VisualAge Pacbase Version 3.5

Vous pouvez nous adresser tout commentaire sur ce document (en indiquant sa référence) via le site Web de notre Support Technique à l'adresse suivante : http://www.ibm.com/software/awdtools/vapacbase/support.html
ou en nous adressant un courrier à :
IBM Paris Laboratory
1, place Jean–Baptiste Clément
93881 Noisy-le-Grand, France.
IBM pourra disposer comme elle l'entendra des informations contenues dans vos commentaires, sans aucune obligation de sa part.

© Copyright International Business Machines Corporation 1983,2007. Tous droits réservés.

Table des Matières

© Copyright IBM Corp. 1983, 2007

© Copyright IBM Corp. 1983, 2007

Notices

Ce document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM. Cela ne signifie pas qu'IBM ait l'intention de les annoncer dans tous les pays où la compagnie est présente.

Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM.

Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

Intellectual Property and Licensing
International Business Machines Corporation
North Castle Drive, Armonk, New-York 10504-1785
USA

Les détenteurs de licences du présent produit souhaitant obtenir des informations sur celui-ci à des fins : (i) d'échange d'informations entre des programmes développés indépendamment et d'autres programmes (y compris celui-ci) et (ii) d'utilisation mutuelle des informations ainsi échangées doivent s'adresser à :

IBM Paris Laboratory
Département SMC
1, place Jean–Baptiste Clément
93881 Noisy-le-Grand
FRANCE

De telles informations peuvent être mises à la disposition du Client et seront soumises aux termes et conditions appropriés, y compris dans certains cas au paiement d'une redevance.

IBM peut modifier ce document, le produit qu'il décrit ou les deux.

Marques

IBM est une marque d'International Business Machines Corporation, Inc.
AIX, AS/400, CICS, CICS/MVS, CICS/VSE, COBOL/2, DB2, IMS, MQSeries, OS/2, VisualAge Pacbase, RACF, RS/6000, SQL/DS et VisualAge sont des marques d'International Business Machines Corporation, Inc. dans certains pays.

Java et toutes les marques et logos incluant Java sont des marques de Sun Microsystems, Inc. dans certains pays.
Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation dans certains pays.
UNIX est une marque enregistrée aux Etats-Unis et/ou dans d'autres pays et utilisée avec l'autorisation exclusive de la société X/Open Company Limited.
D'autres sociétés peuvent être propriétaires des autres marques, noms de produits ou logos qui pourraient apparaître dans ce document.

© Copyright IBM Corp. 1983, 2007

Chapitre 1 : Introduction

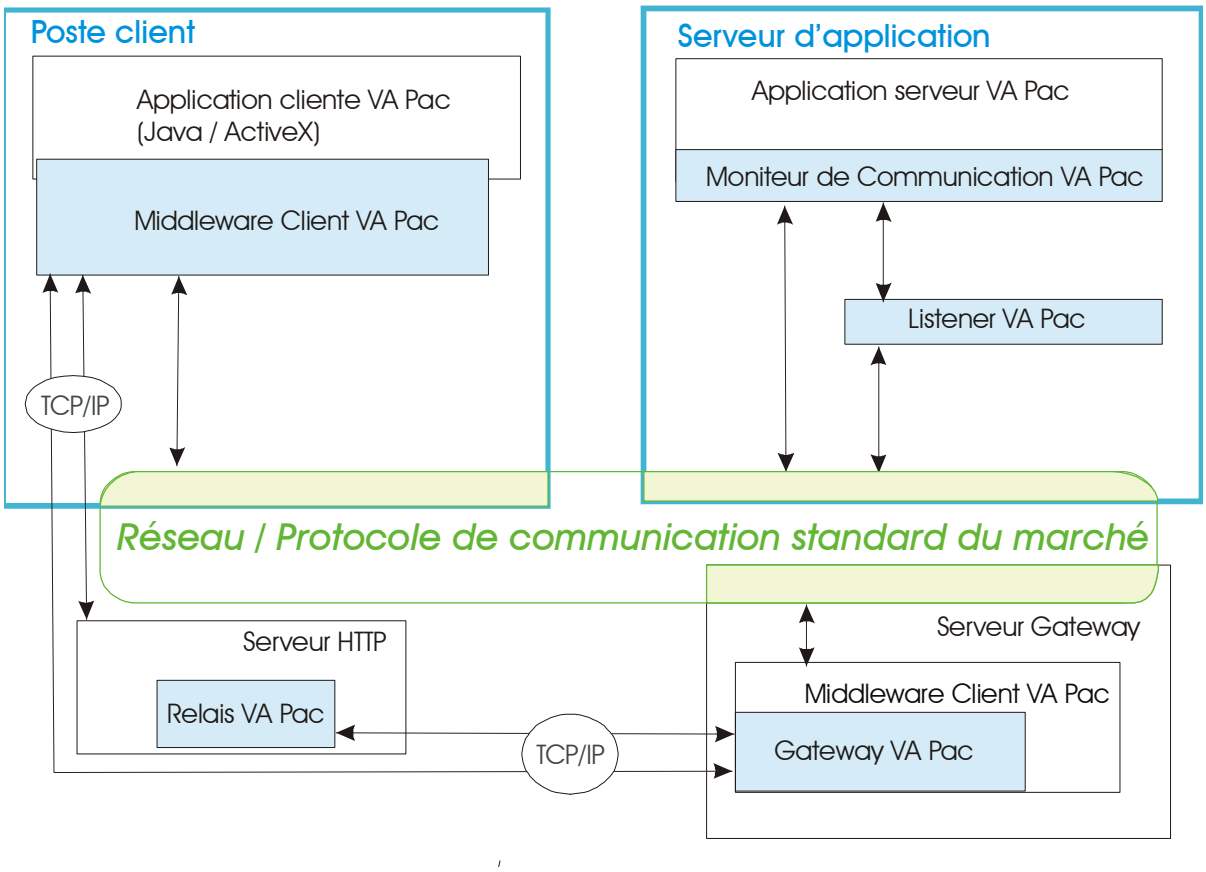
Les fonctions middleware gèrent la communication entre les composants Client et Serveur d'une application utilisant les protocoles de communication du marché.

Vous trouverez dans ce guide toutes les informations relatives à l'exploitation du middleware mis en œuvre pour les applications générées par les modules eBusiness et Pacbase WebConnection.

☞ Les informations sur le middleware utiles au développeur (comme la personnalisation du middleware, le packaging …) sont traitées dans le Manuel "Présentation graphique eBusiness". De plus, le développeur trouvera les méthodes et attributs relatifs au middleware dans le Manuel "Interface publique des composants générés".

Chapitre 2 : Les différents modes de communication

Architecture générale



Le poste client est la machine sur laquelle est installée l'application cliente.
Le middleware client VA Pac est inclus dans le runtime eBusiness de VA Pac.

Communication en mode direct

Dans ce mode de communication, les échanges entre le composant client et le composant serveur sont gérés directement par un protocole de communication du marché (MQ Series, TCP/IP, CICS-ECI, CPI-C…).

La mise en œuvre du protocole de communication choisi est entièrement encapsulée.

En Java, elle est représentée par la classe **`MiddlewareAdapter`**, fournie dans le runtime VA Pac (**`vaprun.jar`**). Cette classe offre une API permettant de définir le contexte de communication avec l'application serveur.

Cette API est documentée dans le Manuel "Interface publique des composants générés".

Sur chaque poste contenant l'application cliente doivent être installés :

- le middleware client VA Pac,
- la partie cliente du protocole de communication utilisé.

La requête du composant client suit alors le chemin suivant : elle est émise par l'application cliente VA Pac (Java ou ActiveX), passe par le middleware client VA Pac, puis est prise en charge par le protocole de communication avant d'aboutir au Moniteur de Communication, qui est le premier composant serveur atteint.

Le chemin jusqu'au moniteur de communication dépend de l'architecture du système de communication (Tuxedo, MQ Series, …) et de la configuration mise en œuvre par l'administrateur du réseau.

Vous pouvez aussi exploiter le module listener Socket VA Pac basé sur un protocole propriétaire du middleware. La requête cliente transite alors par lui avant d'atteindre le Moniteur de Communication. Voir *Chapitre 8 : Utilisation du listener VA Pac*.

Communication via une gateway

La gateway est représentée par le composant **`VapGateway`**.

Il s'agit d'un programme chargé de faire les accès au middleware Client VA Pac qui n'est pas installé sur le poste contenant l'application cliente mais sur un serveur intermédiaire. Les composants client communiquent alors avec la gateway VA Pac qui s'exécute sur ce serveur intermédiaire.

La gateway est donc partagée par toutes les applications clientes et doit donc être lancée en permanence.

Ceci permet de ne pas alourdir le poste où est installée l'application cliente et d'effectuer une gestion optimisée et centralisée des accès serveur.

La requête du composant client suit le chemin suivant : elle est émise par l'application cliente VA Pac (Java ou ActiveX), et va directement dans la gateway, via TCP/IP. Elle passe ensuite par le middleware Client VA Pac installé avec le composant **`VapGateway`**, puis passe par la partie cliente du protocole de communication, puis le réseau lui-même avant d'aboutir au Moniteur de Communication, qui est le premier composant serveur atteint.

Vous pouvez aussi exploiter le module listener Socket VA Pac basé sur un protocole propriétaire du middleware. La requête cliente transite alors par lui avant d'atteindre le Moniteur de Communication. Voir *Chapitre 8 : Utilisation du listener VA Pac*.

Pour tous les protocoles non SOCKET, il faut aussi installer la partie cliente du protocole de communication sur la même machine que celle où est installée la gateway. En revanche, seul le protocole de communication TCP/IP est nécessaire sur chaque poste contenant l'application cliente, et ce protocole est généralement installé par défaut.

Cas particulier : mode relais

Le relais peut être utilisé pour les applications de type Internet. Il permet de décharger le serveur HTTP de la tâche de communication avec les serveurs et de la gestion des contextes des clients connectés. Il est surtout utilisé pour relayer les requêtes arrivant à un serveur HTTP vers une autre machine qui abrite le middleware et la gateway. Ceci allège donc la machine sur laquelle la gateway est installée et l'administrateur n'a pas à gérer les divers protocoles de communication utilisés par ces applications.

Ce composant a également pour fonction d'établir un relais TCP/IP simple (comme un router) permettant à son client de se connecter à une autre adresse TCP/IP qu'il ne pourrait pas atteindre directement (par exemple si elle se trouve derrière un firewall).

Le relais reçoit des requêtes identiques à celles de la gateway et les ré-émet vers une gateway (ou un autre relais). Il se distingue donc de la gateway par le fait qu'il ne fait pas appel au middleware.

Il est représenté par le programme **`vaprelay.jar`**.

Chapitre 3 : Les composants du middleware

Les fichiers listés ci-dessous peuvent avoir une extension **.dll**, **.o**, **.so** ou pas du tout en fonction de la plateforme cible.

Les fichiers avec l'extension **.001** sont les versions avec possibilité d'activer le mode trace pour analyser des problèmes de communication. Pour plus de détails, voir le *Chapitre 10 : Résolution des problèmes de communication*.

Fichiers communs

CharConv.txt	Fichier des tables de conversion codepages
GsComMw	Interface Middleware pour Pacbench/PacDesign (ancienne Station de Travail VA Pac)
GwAdapter	Adapter pour communiquer avec VapGateway
(lib)JavaAdapter	Interface Middleware JAVA/C++ (JNI)
MwAdapter	Adapter pour communication mode Middleware direct
VapUtil	Library des fonctions et classes utilitaires C++
ixomware	Interface générique de la couche de communication IXO
ixomsgen	Libellés d'erreur (Windows seulement)

Interfaces techniques de communication

Ces fichiers regroupent les fonctions d'exécution des services de communication du middleware pour un protocole de communication spécifique.

ixocics	Protocole CICS/ECI
ixocpic	Protocole CPI-C
ixocgtea	Protocole GTEA (GCOS8)
ixoloc	Protocole DLL Cobol locale
ixomqci	Protocole MQSeries-CICS Bridge
ixomqmci	Version pour MQServer
ixomqims	Protocole MQSeries-IMS Bridge
ixomqmim	Version pour MQServer
ixomqs	Protocole MQSeries natif
ixomqm	Version pour MQServer
ixosock	Protocole Socket VAP
ixotims	Protocole IMS Connect
ixotmvs	Protocole Socket CICS
ixottds	Protocole TDS TCP/IP

ixotux Protocole TUXEDO (si non utilisation de **JoltAdapter**)

ixotuxmt Version "Multithreads" pour Natif Client Threads (si non utilisation de **JoltAdapter**

fieldtbl.vap Fichier de conversion FML

Module VAP Gateway

VapGateway Programme VAP Gateway

VapGatewayNT Version installable en service WindowsNT

Pour plus de détails, voir *Communication via une gateway*.

Module VAP Relay

vaprelay.jar Package des classes Java de VAP Relay.

Ce module est autonome et ne nécessite donc aucun autre composant du package Middleware.

Le fichier **.jar** contient :

- le package **com.ibm.vap.relay**,
- un fichier **readme.txt**,
- un script DOS de lancement du relais : **vaprelay.bat**,
- un script Unix de lancement du relais : **vaprelay**.

Pour plus de détails, voir *Cas particulier : mode relais*.

Module VAP Listener

listener Programme VAP Listener

Sous Unix, il est livré seulement sous forme d'un objet compilé.

Avant de pouvoir l'utiliser, il faudra faire une édition de liens avec les autres objets compilés de l'application serveur.

Un exemple de makefile est fourni dans le package.

dial Agent de dialogue avec client (Windows seulement)

CodePageConv Fonctions de conversion de codepages

VapUtil1 Library des fonctions et classes utilitaires C++ (Unix seulement)

Pour plus de détails, voir *Chapitre 8 : Utilisation du listener VA Pac*

Chapitre 4 : Protocoles disponibles par plateformes Serveur

Précisions importantes sur les plateformes clientes ou serveurs gateway indiqués dans les tableaux :

- **Windows** : Système Windows 32 bits (98, NT, 2000, XP).
- **AIX** : à partir de la v4.2. Minimum V4.3 pour GCOS7-TDS-TCP/IP et GCOS8-GTEA.
- **Solaris** : à partir de la v2.5.1.
- **Unix OS390** : Unix System Services for OS390, V2R9.
- **Linux** : Debian GNU/Linux 2.2.19 avec glibc 2.2.4-7.

OS/390 - CICS

Protocoles / Plateforme cliente	Windows	AIX	Solaris	Unix OS390	Linux
CPI-C	Oui				
ECI	Oui				
EXCI				Oui	
TCP/IP Socket	Oui	Oui	Oui	Oui	Oui
MQSeries	Oui	Oui	Oui		Oui
MQ-CICS Bridge	Oui	Oui	Oui		Oui

OS/390 - IMS

Protocoles / Plateforme cliente	Windows	AIX	Solaris	Unix OS390	Linux
CPI-C	Oui				
IMS Connect	Oui	Oui	Oui	Oui	Oui
MQSeries	Oui	Oui	Oui		Oui
MQ-IMS Bridge	Oui	Oui	Oui		Oui

DOS/VSE – CICS

Protocoles / Plateforme cliente	Windows	AIX	Solaris	Unix OS390	Linux
CPI-C	Oui				

AIX

Protocoles / Plateforme cliente	Windows	AIX	Solaris	Unix OS390	Linux
ECI	Oui				
TCP/IP Socket	Oui	Oui	Oui		Oui
MQSeries	Oui	Oui	Oui		Oui
Tuxedo	Oui	Oui			

Solaris

Protocoles / Plateforme cliente	Windows	AIX	Solaris	Unix OS390	Linux
TCP/IP Socket	Oui	Oui	Oui		Oui
MQSeries	Oui	Oui	Oui		Oui
Tuxedo	Oui	Oui			

Windows

Protocoles / Plateforme cliente	Windows	AIX	Solaris	Unix OS390	Linux
CPI-C (TX-Series)	Oui				
ECI (TX-Series)	Oui				
TCP/IP Socket	Oui	Oui	Oui		Oui
MQSeries	Oui	Oui	Oui		Oui
Tuxedo	Oui	Oui			

AS400

Protocoles / Plateforme cliente	Windows	AIX	Solaris	Unix OS390	Linux
TCP/IP Socket	Oui	Oui	Oui		Oui

Compaq OpenVMS

Protocoles / Plateforme cliente	Windows	AIX	Solaris	Unix OS390	Linux
TCP/IP Socket	Oui	Oui	Oui		Oui

Compaq Tru64 UNIX

Protocoles / Plateforme cliente	Windows	AIX	Solaris	Unix OS390	Linux
TCP/IP Socket	Oui	Oui	Oui		Oui

HP-UX

Protocoles / Plateforme cliente	Windows	AIX	Solaris	Unix OS390	Linux
TCP/IP Socket	Oui	Oui	Oui		Oui

Linux

Protocoles / Plateforme cliente	Windows	AIX	Solaris	Unix OS390	Linux
TCP/IP Socket	Oui	Oui	Oui		Oui

NUMA-Q DYNX (SEQUENT)

Protocoles / Plateforme cliente	Windows	AIX	Solaris	Unix OS390	Linux
TCP/IP Socket	Oui	Oui	Oui		Oui

Host Tandem OSS-Guardian (Himalaya)

Protocoles / Plateforme cliente	Windows	AIX	Solaris	Unix OS390	Linux
TCP/IP Socket	Oui	Oui	Oui		Oui
Tuxedo	Oui	Oui			

Host GCOS7 - TDS

Protocoles / Plateforme cliente	Windows	AIX	Solaris	Unix OS390	Linux
TDS-TCP/IP	Oui	Oui			

Host GCOS8 - TP8

Protocoles / Plateforme cliente	Windows	AIX	Solaris	Unix OS390	Linux
GTEA	Oui	Oui			

Chapitre 5 : Notion de localisation

Pour définir la communication, deux notions sont fondamentales :

- la notion de localisation, qui décrit le contexte d'utilisation du middleware,
- le mode de communication : il peut être direct ou via une gateway (avec éventuellement un relais).

Chaque localisation :

- Identifie le protocole utilisé pour accéder au serveur VisualAge Pacbase,
- Donne les coordonnées physiques de ce serveur pour ce protocole,
- Définit les paramètres de communication nécessaires au bon fonctionnement du protocole.

Les localisations sont regroupées dans le fichier **`vaplocat.ini`**. Le code de chaque localisation est indiqué dans le fichier **`vaplocat.ini`** entre <…>. A l'intérieur de chaque localisation sont spécifiés des paramètres de communication, dont la syntaxe est la suivante : **`nom du paramètre=valeur du paramètre`**.

Il existe deux types de paramètres :

- des paramètres relatifs aux fonctions générales de communication (type de la communication, longueur du message…),
- des paramètres spécifiques à chaque type de communication (nom du projet pour TCP/TDS, nom du manager de queue pour MQSERIES …). Ces paramètres sont préfixés par **`IXO_`**. Le couple nom/valeur du paramètre est transmis tel quel à la couche middleware spécifique à la communication.

Vous paramétrez le fichier **`vaplocat.ini`** en utilisant l'outil Editeur de Localisations (disponible depuis le module eBusiness du Developer workbench, ou indépendamment). Voir l'aide en ligne de cet outil.

La liste des paramètres ainsi que leur signification est documentée au *Chapitre 9 : Description & Configuration des protocoles*, dans la section *Configuration* de chaque protocole.

Chapitre 6 : Utilisation de VapGateway

Lancement

Pour lancer la gateway, positionnez-vous sur le répertoire contenant **VapGateway** et tapez la commande suivante (l'ordre des options est sans importance) :

```
VapGateway [-h]
-s|i|d TCP_PORT_NUM [-l LOCATION_FILE] [-c CHAR_CONV_FILE]
[-t TRACE_LEVEL] [-tf TRACE_FILE]
    [-min NB_MIN_CONNECTION] [-max NB_MAX_CONNECTION]
[-clean CLEANING_INTERVAL] [-clientTimeout CLIENT_TIMEOUT]
[-connectionTimeout CONNECTION_TIMEOUT]
[-retry RETRY_INTERVAL]
```

avec :

- **-h** : option qui permet d'afficher la syntaxe
- **-s** : lance directement la gateway
- **-i** : installe la gateway comme un service Windows NT

  Par défaut, la gateway est installée en mode de démarrage automatique.
- **-d** : désinstalle la gateway des services Windows NT
- **TCP_PORT_NUM** : positionne le port de communication utilisé par les objets Proxy. La valeur par défaut est 5647.
- **-l LOCATION_FILE** : spécifie le fichier des localisations. Le fichier par défaut est **VAPLOCAT.INI** dans le répertoire de l'exécutable **VapGateway**.
- **-c CHAR_CONV_FILE** : spécifie le fichier de conversion du code page

  Le fichier par défaut est **CHARCONV.TXT** dans le répertoire de l'exécutable **VapGateway**.
- **-t TRACE_LEVEL** : positionne le niveau de trace.
  - 0 : aucune trace
  - 1 : traces d'erreurs (défaut)
  - 3 : traces générales
  - 5 : traces détaillées
- **-tf TRACE_FILE** : Indique le fichier trace.

  Le répertoire par défaut est **VAPTRACE** dans le répertoire de l'exécutable VapGateway.
- **-min NB_MIN_CONNECTION** : Spécifie le nombre minimum de connexions serveur maintenues ouvertes dans le pool de connexions lors d'un nettoyage ('cleaning').
  Les connexions sont ouvertes au fur et à mesure des besoins. Lorsque le nombre minimum de connexions est atteint, il est maintenu et le mécanisme de 'cleaning' est mis en œuvre.

  La valeur par défaut est 0..

- **-max NB_MAX_CONNECTION** : Spécifie le nombre maximum de connexions serveur dans le pool de connexions.
  Avant de créer une nouvelle connexion, la gateway contrôle le nombre maximum de connexions. S'il n'est pas atteint, une nouvelle connexion est créée. S'il est atteint, la connexion inoccupée (paramètre **clientTimeout**) la plus récente est détruite afin de créer une nouvelle connexion. Si toutes les connexions sont utilisées, la demande de connexion est bloquée jusqu'à ce qu'une connexion se libère ou que le délai de **connection Timeout** soit atteint. Ce paramètre a un impact important sur les performances.

  La valeur par défaut est illimitée.
- **-clean CLEANING_INTERVAL** : Spécifie le nombre de secondes entre deux nettoyages de connexions serveur inactives dans le pool de connexions.
  Pour des raisons de performance, la couche middleware gère un groupe de connexions serveur associées. Une connexion inactive est une connexion, dans le groupe, qui n'a pas été utilisée depuis le dernier nettoyage.
  Vous devez affecter une petite valeur à ce paramètre (1 seconde par exemple) si vous ne souhaitez pas conserver de connexions au serveur qui soient inutilisées (afin de limiter l'utilisation des ressources). En revanche, vous devez affecter une valeur importante à ce paramètre (60 secondes par exemple) si vous souhaitez obtenir de meilleures performances (réduire le nombre de connexions / déconnexions / reconnexions) .
  La valeur du **Cleaning_Interval** ne doit pas être supérieure au temps limite de connexion de la partie serveur, sinon, des connexions actives au niveau du pool seront conservées inutilement.

  La valeur par défaut est 60.
- **-clientTimeout CLIENT_TIMEOUT** : Spécifie le temps de maintien des connexions clientes restant inactives.

  La valeur par défaut est 120.
- **-connectionTimeout CONNECTION_TIMEOUT** : Spécifie le temps maximum d'attente (en secondes) d'un client pour obtenir une connexion serveur, quand le nombre maximum de connexions serveur dans le pool est atteint.

  La valeur par défaut est illimitée.
- **-retry RETRY_INTERVAL** : Spécifie le nombre de secondes entre deux essais de communication vers le Serveur en cas de réception d'une erreur de communication..
  Une tentative de reconnexion est effectuée uniquement en cas d'erreur de communication sur une connexion restée inactive pendant le nombre de secondes du dernier **RETRY_INTERVAL**.
  La valeur par défaut est **0**.
  Vous pouvez inhiber ce mécanisme en donnant la valeur **-1**.
- **-pcv** : Inutilisé. Conservé pour comptabilité uniquement. Dans cette version de VapGateway, cette option est toujours active.

Identification de la version de VapGateway

Vous pouvez identifier la version de **VapGateway** en positionnant le niveau de trace à **1** minimum. L'identification de la version se trouve dans l'en-tête du fichier trace.

Voici un exemple de l'en-tête d'un fichier trace obtenu lors du lancement de la gateway en interactif, avec un niveau de trace égal à **1** :

```
[VapGateway 17:51:28:94
     VisualAge Pacbase (*) v3.5 VapGateway vm350v01
     Licensed Materials - Property of IBM 5655-F37
     © Copyright IBM Corp. 1983, 2006. All Rights Reserved]
```

Dans cet exemple, **vm350v01** est l'identifiant de la version.

Sous Windows, vous pouvez identifier plus facilement la version grâce à la fiche de propriétés du fichier **VapGateway.exe**. Cette fiche est consultable via l'explorateur Windows.

Pour plus d'informations sur la trace, consultez le *Chapitre 10 : Résolution des problèmes de communication*.

Chapitre 7 : Utilisation de vaprelay.jar

Le **vaprelay** est un composant Java autonome exécutable avec un runtime Java (version 1.2 ou plus conseillée).

Pour pouvoir fonctionner, le relais doit connaître deux informations sur la gateway avec laquelle il communique : son nom ou adresse IP et son port d'écoute. Il doit d'autre part se mettre à l'écoute des clients.

Un exemple de lancement du programme relais est fourni sous forme de deux fichiers de commandes contenus dans **vaprelay.jar** :

- **vaprelay.bat** pour Windows,
- **vaprelay** pour Unix.

Vous devez extraire ces fichiers et les adapter à vos besoins.

Le runtime Java doit être installé au préalable et n'est pas fourni dans ce package.

En standard, vous lancez le relais avec la commande suivante :

```
vaprelay <delegateHostName> [delegatePort [listeningPort]]
```

avec :

- **delegateHostName** : nom ou adresse IP du serveur sur lequel la gateway s'exécute.
- **delegatePort** : numéro de port TCP/IP sur lequel la gateway écoute (**5647** par défaut).
- **listeningPort** : numéro de port TCP/IP sur lequel le relais écoute les clients (**5647** par défaut).

Chapitre 8 : Utilisation du listener VA Pac

Si vous avez déjà VisualAge Pacbase, le listener vous permet d'utiliser le protocole propriétaire Socket VA Pac pour faire communiquer l'application eBusiness générée (Java ou ActiveX) avec la partie serveur VA Pac. Ceci vous évite d'installer un système de communication. Le protocole Socket VA Pac s'appuie sur la couche de transport TCP/IP du réseau de communication.

Pour installer le listener, copiez dans un répertoire les deux exécutables suivants :

- **BvpServer.exe**
- **BvpDial.exe**. Il existe deux versions de cet exécutable, correspondant au compilateur Cobol utilisé : Acucobol ou Microfocus. La version Acucobol est située dans le répertoire **ACU** du listener et la version Microfocus dans le répertoire **MF**.

N'oubliez pas d'ajouter ce répertoire d'installation dans la variable **PATH**.

Le listener est livré pour les plateformes suivantes :

- AIX,
- AS400,
- HPUX,
- Linux,
- OSF1,
- Open VMS.
- Solaris,
- Tandem,
- Unix OS390,
- Windows.

Lancement du listener

Vous lancez le listener sur le serveur d'application, via la commande suivante :

- Pour Windows :

  ```
  BvpServer [-h] [-i|d|s <Port_Number>
  [Environment_File][Security_program]]
  ```

- Pour toute autre plateforme :

  ```
  Listener [-h] [-s <Port_Number>
  [Environment_File][Security_program]]
  ```

avec :

- **-h**: message d'aide
- **-i<Port_Number>[Environment_File][Security_Program]** : installer le listener comme service Windows NT
- **-d <Port_Number>**: désinstaller le service listener
- **-s <Port_Number>**: lancement direct du listener
- **<Port_Number>**: Valeur décimale du port TCP/IP
- **<Environment_File>**: fichier de positionnement des variables d'environnement
- **<Security_Program>**: programme appelé par le listener pour le contrôle de sécurité

Pour les plateformes autres que Windows, l'exécutable du listener doit être fabriqué avant de pouvoir être utilisé. Ce programme nécessite une édition de liens avec des bibliothèques de la Base de Données de votre application. Référez-vous au fichier **readme** fourni dans le package middleware destiné à votre plateforme cible.

Spécificités de l'utilisation du listener sous UNIX

Pour la plateforme UNIX, vous devez transférer :

- les modules **bvpserver.o**, **bvppause.o**, **config.**o, **dtime.o**, **environ.**o, **general.o**, **lockdb.o**, **mems.o**, **sems.o**, **standard.o** et **tn3270.o**,
- le fichier **Makefile** et
- les bibliothèques **CodePageConv.so** et **VapUtil1.so**

sur le serveur UNIX avant l'édition des liens via la commande :

make -f Makefile.

☞ Les bibliothèques **CodePageConv.so** et **VapUtil1.so** doivent se trouver dans le répertoire supérieur pendant la fabrication des modules.

Le résultat est le fichier **listener.exe**, situé lui-aussi dans le répertoire supérieur. Vous pouvez renommer cet exécutable en '**server**' par exemple. Dans ce cas, vous le lancerez via la commande :

server path port [-t <timeout>]&

avec:

- **path**: chemin des programmes exécutables des serveurs,
- **port**: numéro de socket utilisé par le listener
- **timeout**: délai d'attente maximum pour une connexion sans données ('0' par défaut pour un délai illimité)

Vous pouvez appliquer différents niveaux de trace :

- Niveau 1 : trace minimum des traitements du listener
- Niveau 2 : trace détaillée des traitements du listener
- Niveau 4 : trace des messages échangés entre le listener et le poste client.

Pour activer une trace, vous devez redémarrer le listener après avoir initialisé la variable d'environnement **SRV_TRACE**. Par exemple :

- **export SRV_TRACE=1** pour une trace de niveau 1
- **export SRV_TRACE=3** pour une trace de niveau 1 et 2
- **export SRV_TRACE=5** pour une trace de niveau 1 et 4

Pendant l'utilisation du mode debug, les traces s'écrivent dans les fichiers **srv<process pid>.txt** et **dial<process pid>.txt** situés dans le répertoire **/tmp**.

La variable d'environnement **SRV_DIR** vous permet d'écrire ces traces dans un autre repertoire que **/tmp**. Par exemple :

export SRV_DIR=$HOME/tmp

Chapitre 9 : Description & Configuration des protocoles

IMS Connect

Prérequis

IMS V6

IMS Connect V1R1

Cinématique des échanges

- Echange standard

```
REQUETE CLIENTE        FLOT                    REQUETE IMS CONNECT
SEND---------------->IRM/TRAN/DATA ------------>RECEIVE
RECEIVE<---------------LLzz/DATA<---------------SEND
...                      ...                      ...
RECEIVE<---------------LLzz/DATA<---------------SEND
RECEIVE<------------------CSM<------------------SEND
```

La connexion reste active en mode **SOCKET PERSISTENT** ; elle doit être explicitement fermée à la fin de l'échange.

- Erreur détectée par IMS Connect ou le message exit

```
REQUETE CLIENTE  FLOT                   REQUETE IMS CONNECT
SEND---------------->IRM/TRAN/DATA ------------>RECEIVE
RECEIVE<------------------RSM<------------------SEND
                Connexion fermée
```

Description du fonctionnement

Le "user message exit" utilisé est le HWSIMSO0 en configuration suivante :

- Mode conversationnel
- Commit mode et Sync level définis par défaut dans HWSIMSO0 (**commit mode 1**, et **sync level NONE)**

Configuration

La configuration du protocole de communication passe par la déclaration de paramètres dans le fichier des localisations **vaplocat.ini**. (pour des explications sur ce fichier, voir le *Chapitre 5 : Notion de localisation*).

Les paramètres de la partie cliente de ce protocole de communication sont documentés à l'*Annexe : Paramétrage des logiciels externes*.

Paramètres obligatoires

COMM_TYPE=TCPIMS

MONITOR

Nom du programme Moniteur de Communication VA Pac.

Ce Moniteur de Communication doit être généré avec le type **SOCKET**.

MESSAGE_LENGTH

Longueur maximum du message attendu par le Moniteur de Communication.

IXO_TRANSID

Nom de la transaction du Moniteur de Communication. (8 car.).

IXO_ADDRESS

Adresse IP ou nom logique du host suivi du numéro de port d'IMS Connect, séparé par un caractère ESPACE. (30 car. max.)

IXO_DATASTORE

Nom du lien vers IMS défini dans IMS Connect (8 car. max.)

IXO_RACFGROUP

Nom du groupe RACF pour IMS Connect. (8 car. max.)

Paramètres facultatifs

HOST_ENCODING

Valeur du code page serveur. Cette valeur doit être définie dans le fichier des tables de conversion code page (**charconv.txt**) fourni dans le package middleware.

IXO_TIMEOUT

Temps d'attente de réception d'une réponse, exprimé en secondes (30 secondes par défaut).

IXO_PERSISTENT

Mode de persistance de la connexion Socket.
Y persistant (mode par défaut)
N non persistant

MQSeries

Prérequis

MQSeries 5.2

Description du fonctionnement

MQSeries est un système de communication basé sur les échanges de messages via des files de messages (Queue). La gestion de ces échanges est prise en charge par le gestionnaire MQSeries (Queue Manager).

Le Middleware MQSeries utilise 2 files de messages :

- la file de requête (Request Queue)

  Cette Queue est utilisée pour envoyer les messages de requête à l'application Serveur.

- la file de réponse (Reply Queue)

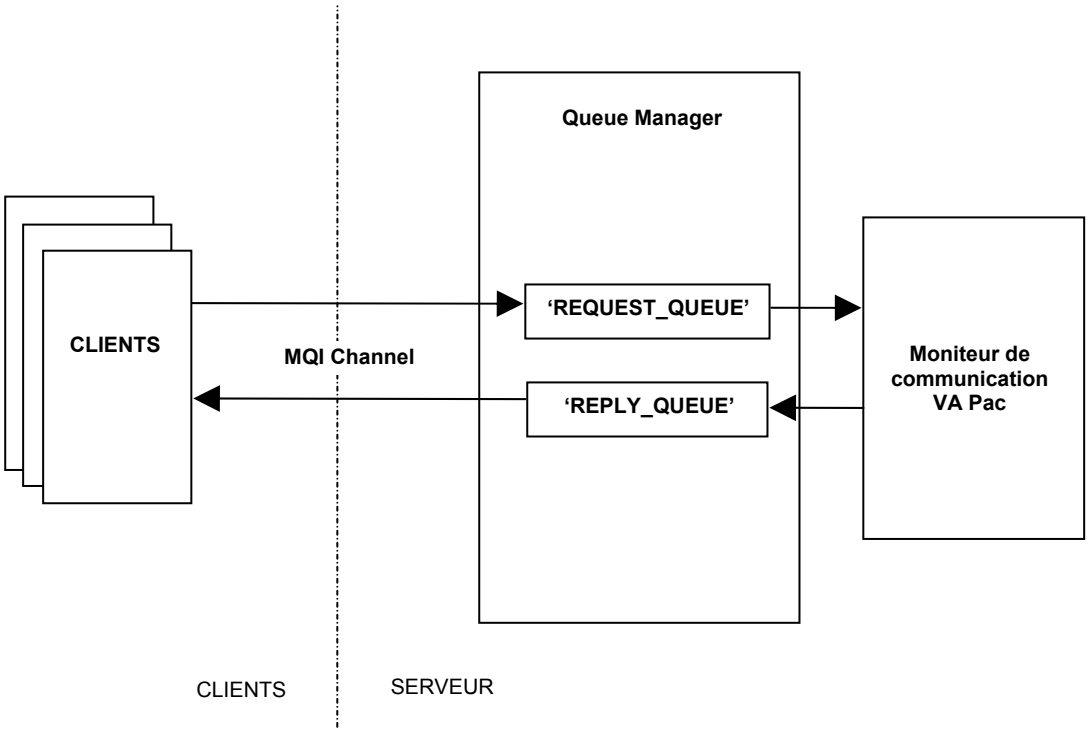
  C'est dans cette Queue que l'application Client reçoit les réponses du serveur. Son nom est transmis à l'application serveur dans la partie en-tête du message de requête.

Les messages de requête déposés dans la Request Queue sont lus par le Moniteur de Communication MQSeries sur le serveur, puis sont traités selon le même principe que pour les autres types de système de communication. Les réponses retournées par les programmes serveurs sont transmises au client via la Reply Queue selon le nom reçu dans la partie en-tête de chaque message de requête. Pour créer le lien entre requête et réponse, on affecte aux messages de réponse le numéro d'identification du message de requête correspondant.

Définition des objets MQSeries

Architecture Client/Serveur

Pour ce type d'architecture, l'application Client et l'application Serveur partagent le même Queue Manager. Celui-ci ainsi que tous les objets MQSeries sont implémentés sur le Serveur de l'application.



Dans ce contexte, les objets MQSeries à définir sont:

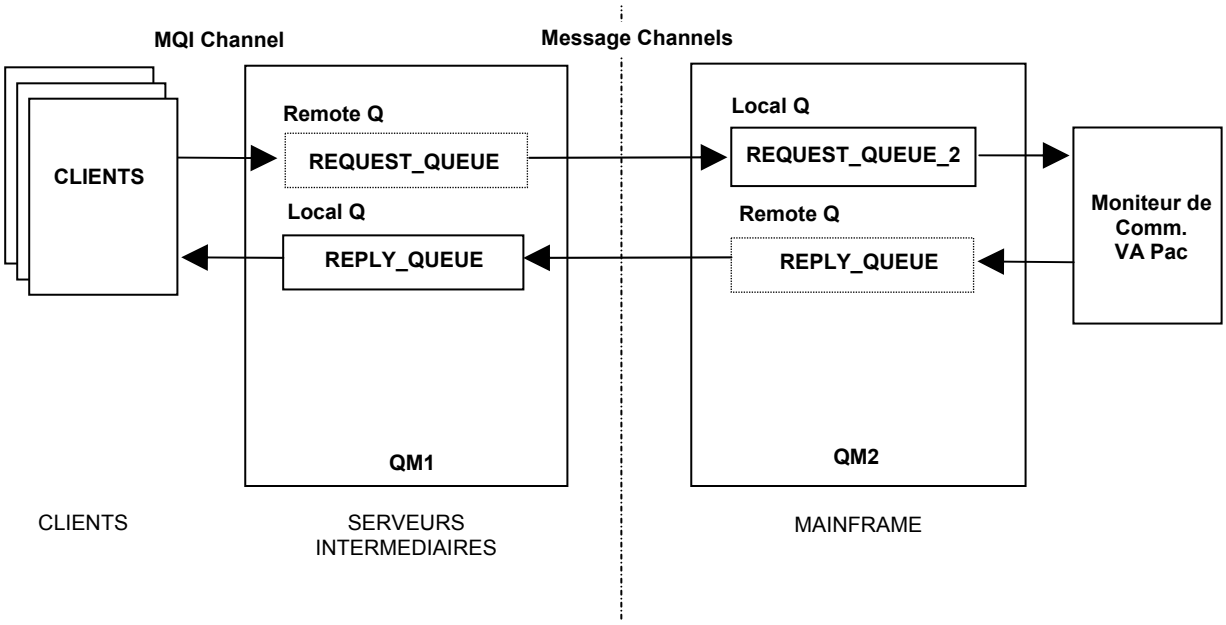
- 2 **`Local Queue`**
  - **`Request Queue`**
  - **`Reply Queue`**

- 1 **`MQI Channel SVRCONN/CLNTCONN`**

Architecture Distribuée

Dans ce cas, plusieurs Queue Managers sont interconnectés pour former un réseau MQSeries.

Chacune des applications, Client et Serveur, est connectée à son propre Queue Manager dit «Local ».



Les objets MQSeries à définir sont :

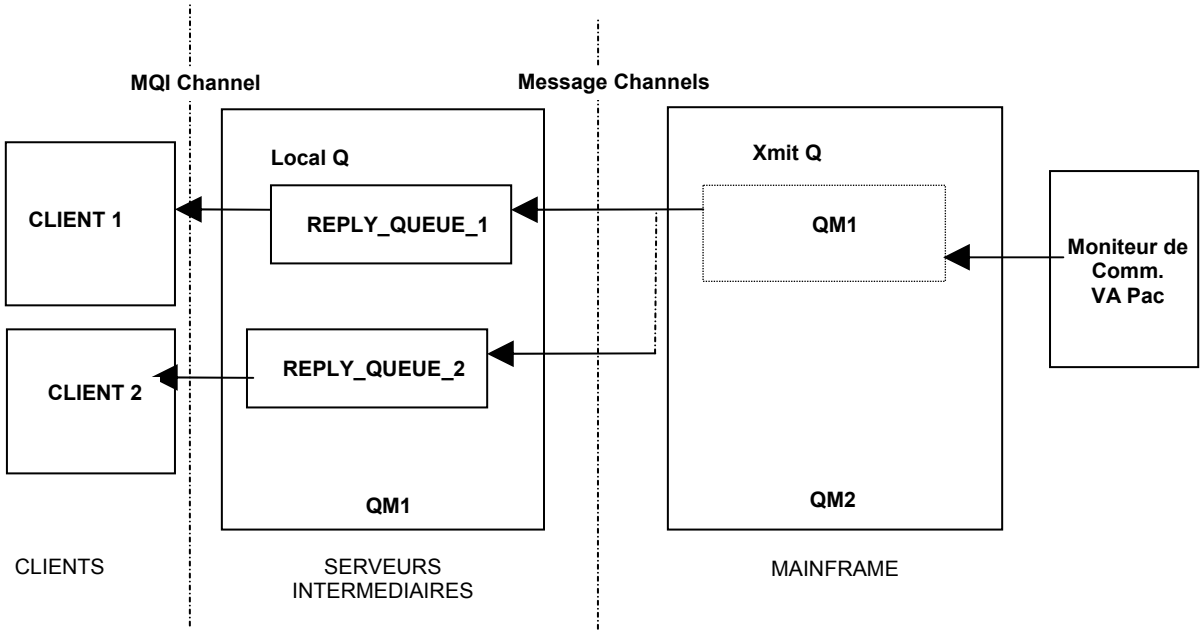
- Pour le Queue Manager local du Client
  1 `Local Queue`: `Reply Queue`
  1 `Remote Queue` : `Request Queue`
  1 `MQI Channel` `SVRCONN/CLNTCONN`
  2 `Message Channel` : 1 `SDR` + 1 `RCVR`

- Pour le Queue Manager côté Serveur:
  1 `Local Queue`: `Request Queue`
  1 `Remote Queue`: `Reply Queue` (de même nom que la `Reply Queue` du Client)
  2 `Message Channel` : 1 `SDR` + 1 `RCVR`

Ne sont pas cités les objets techniques internes au système MQSeries, tels que Transmit Queue, Dead Letter Queue,...

Une Remote Queue n'est pas réellement une Queue de message ; sa définition permet uniquement au Queue Manager de connaître les références nécessaires (Nom du Queue Manager et de la Queue distante, Transmit Queue) pour le routage des messages vers le Queue destinataire.
L'obligation d'utiliser des Remote Queue s'explique par le fait qu'une application ne peut lire que dans une Local Queue appartenant à son propre Queue Manager.

Utilisation de la fonction Name Resolution de MQSeries :



Cette technique permet de réduire assez considérablement le nombre d'objets MQSeries. Ainsi, quelque soit le nombre de Reply Queues nécessaires aux applications Client, seul le représentant suivant côté serveur est nécessaire :

1 **`Transmit Queue`**: de même nom que le Queue Manager du Client

Configuration

La configuration du protocole de communication passe par la déclaration de paramètres dans le fichier des localisations **`vaplocat.ini`**. (pour des explications sur ce fichier, voir le *Chapitre 5 : Notion de localisation*).

Les paramètres de la partie cliente de ce protocole de communication sont documentés à l'*Annexe : Paramétrage des logiciels externes*.

Paramètres obligatoires

`COMM_TYPE=MQSERIES` ou

`COMM_TYPE=MQSERVER` si le middleware est installé sur la même machine que MQSeries Server et que l'on souhaite utiliser un lien direct avec MQSeries.

`MONITOR`

Nom du programme Moniteur de Communication VA Pac.

Ce Moniteur de Communication doit être généré avec le type **`MQSERIES`**.

`MESSAGE_LENGTH`

Longueur maximum du message attendu par le Moniteur de Communication.

`IXO_QUEUEMANAGER`

Nom du Queue Manager local de l'application Cliente (48 car. max).

IXO_REQUESTQUEUE

Nom de la Queue des messages de requête (48 car. max).

IXO_REPLYQUEUE

Nom de la Queue des messages de réponse (48 car. max).

Paramètres facultatifs

HOST_ENCODING

Valeur du code page serveur. Cette valeur doit être définie dans le fichier des tables de conversion code page (**charconv.txt**) fourni dans le package middleware.

IXO_TIMEOUT

Temps d'attente de réception d'une réponse, exprimé en secondes (30 secondes par défaut).

IXO_DYNAMICREPLYQUEUE

Lorsque ce paramètre est renseigné, la Queue de réponse est créée dynamiquement par MQSeries pour pouvoir transmettre les messages de réponse (cf. Documentation MQSeries pour l'utilisation de ce type de Queue).

IXO_REQUESTEXPIRY

Délai d'expiration d'une requête, exprimé en secondes.
C'est la durée au delà de laquelle une requête (réponse incluse) est considérée comme perdue. Tous les messages relatifs à cette requête seront alors automatiquement purgés par le gestionnaire MQSeries.
Infini par défaut.

Moniteur de Communication VA Pac

Vous générez ce Moniteur dans VA Pac en spécifiant le type de communication **MQSERIES**.

Le Moniteur peut être déclenché manuellement ou automatiquement (TRIGGER MQSeries) et en mode batch ou transactionnel selon l'environnement de l'application Serveur.

Deux paramètres en entrée doivent être passés lors de son lancement : le nom de la **Request Queue** suivi du nom du **Queue Manager** , séparés par un espace. Dans le cas de l'utilisation du Trigger pour déclencher automatiquement le Moniteur de Communication, ces deux paramètres doivent être transmis via la zone **USER DATA** de l'objet **Process Definition** associé au **Request Queue**.

Les options de génération VA Pac sont les suivantes :

WAITINT

Temps d'attente (en secondes) de l'arrivée d'un message suite pour une requête multi-messages (1 seconde par défaut).

WAITINT1

Intervalle de temps (en secondes) durant laquelle le Moniteur reste à l'écoute sur sa **request queue**.

Illimité par défaut.

MQSeries - CICS Bridge

Prérequis

MQSeries 5.2

Description du fonctionnement

Côté serveur, le module CICS Bridge fonctionne entre MQSeries et le moniteur de communication VA Pac. Il communique avec l'application cliente via des objets MQSeries, et avec le Moniteur de Communication VA Pac via la commande **EXEC CICS LINK**.

Le protocole défini entre le composant Client et le Moniteur de Communication est similaire à celui implémenté pour les autres systèmes de communication synchrone (CPI-C ou ECI):

Quand le message de requête a une longueur inférieure à la limite physique définie par **MESSAGE_LENGTH** dans le fichier de paramètres (**vaplocat.ini**), il est envoyé en une seule fois. Sinon, il est découpé en segments de message. La partie Client envoie le premier segment et attend que le Moniteur de Communication VA Pac lui retourne un accusé de réception pour émettre le suivant, et ainsi de suite jusqu'au dernier segment du message.

Le Moniteur de Communication reçoit chaque segment de message directement dans la **COMMAREA**. Après réception du dernier segment, il reconstitue le message logique complet puis traite la requête. De la même manière que pour l'émission du message de requête, si la longueur du message de réponse est inférieure à la longueur maximale définie, la réponse est retournée au Client en une seule fois dans la **COMMAREA**. Sinon, le Moniteur applique le même principe de découpage et d'envoi de segments que pour le message de requête.

Structure des messages de requête :

<MQMD><MQCIH><TransID><COMMAREA data>

Structure des messages de réponse :

<MQMD><MQCIH><TransID><COMMAREA data>

Structure des messages d'erreur renvoyés par le Bridge :

<MQMD><MQCIH><ErrorText>

MQMD

Descripteur du message de requête :

- **CorrelId = MQCI_NEW_SESSION**
- **MsgType = MQMT_REQUEST**
- **Format = MQFMT_CICS**
- **Report=MQRO_EXCEPTION+MQRO_EXPIRATION+MQRO_PASS_MSG_ID + MQRO_COPY_MSG_ID_TO_CORREL_ID+MQRO_DEAD_LETTER_Q;**
- **Expiry** = paramètre externe IXO
- **CodedCharSetId** = paramètre externe IXO

MQCIH

En-tête du Bridge CICS du message de requête :

- **Version** = paramètre externe IXO
- **Format = MQFMT_STRING**
- **ReplyToFormat = MQFMT_STRING**
- **UOWControl = MQCUOWC_ONLY**
- **LinkType = MQCLT_PROGRAM** (DPL program)
- **Authenticator** = mot de passe
- **TransactionId** = code transaction du Moniteur de Communication

TransID

Nom de la transaction CICS du Moniteur de Communication.

COMMAREA

Segment de message applicatif transmis dans la COMMAREA.

ErrorText

Libellé d'erreur lorsque le code retour dans l'en-tête **MQCIH** (**ReturnCode**) indique une erreur.

D'autres champs de l'en-tête **MQCIH** donnent également des informations sur l'erreur rencontré :

- **CompCode**
- **Reason**
- **Function**
- **AbendCode**

Définition des objets MQSeries

Les objets MQSeries sont documentés au point *Définition des objets MQSeries* pour le protocole MQSeries.

On peut définir un Trigger (FIRST) au niveau de la Request Queue pour démarrer automatiquement le Bridge Monitor.

Configuration

La configuration du protocole de communication passe par la déclaration de paramètres dans le fichier des localisations **vaplocat.ini**. (pour des explications sur ce fichier, voir le *Chapitre 5 : Notion de localisation*).

Les paramètres de la partie cliente de ce protocole de communication sont documentés à l'*Annexe : Paramétrage des logiciels externes*.

Paramètres obligatoires

COMM_TYPE=MQCICS ou
COMM_TYPE=MQMCICS si le middleware est installé sur la même machine que MQSeries Server et que l'on souhaite utiliser un lien direct avec MQSeries.

MONITOR

Nom du programme Moniteur de Communication VA Pac.

Ce Moniteur de Communication doit être généré avec le type **MQBRIDGE**.

MESSAGE_LENGTH

Longueur maximum du message attendu par le Moniteur de Communication.

IXO_TRANSID

Nom de la transaction du Moniteur de Communication (8 car.).

IXO_QUEUEMANAGER

Nom du Queue Manager local (48 car. max.)

IXO_REQUESTQUEUE

Nom de la Queue des messages de requête (48 car. max.)

IXO_REPLYQUEUE

Nom de la Queue des messages de réponse (48 car. max.)

Paramètres facultatifs

IXO_TIMEOUT

Temps d'attente de réception d'une réponse, exprimé en secondes (30 secondes par défaut).

IXO_DYNAMICREPLYQUEUE

Lorsque ce paramètre est renseigné, la Queue de réponse est créée dynamiquement par MQSeries pour pouvoir transmettre les messages de réponse (cf. documentation MQSeries pour l'utilisation de ce type de Queue). Ce paramètre doit alors contenir le nom dynamique de la Queue (Dynamic Queue name) et il faut fournir le nom du "Queue Model" dans le paramètre (48 car. max.)

IXO_REQUESTEXPIRY

Délai d'expiration du message de requête, exprimé en secondes (9 car. max.)

IXO_HEADERVERSION

Version de la structure **MQCIH** : 1(défaut) ou 2, en fonction de l'interface CICS-Bridge utilisée (1 car. max.)

IXO_LOCALCCSID

Code du jeu de caractères de la machine en local (819 par défaut) (9 car. max.)

Ne pas définir **HOST_ENCODING**, car le transcodage est pris en charge par MQSeries.

Pour l'architecture distribuée DQM, les Channels émetteurs doivent avoir l'option **CONVERT** positionnée à **YES**.

MQSeries - IMS Bridge

Prérequis

MQSeries 5.2

Description du fonctionnement

Côté serveur, le module IMS Bridge fonctionne entre MQSeries et le moniteur de communication VA Pac. Il communique avec l'application cliente via des objets MQSeries, et avec le Moniteur de Communication VA Pac via l'interface OTMA.

Le protocole défini entre le composant Client et le Moniteur de Communication est similaire à celui implémenté pour les autres systèmes de communication synchrone (CPI-C ou ECI).

Quand le message de requête a une longueur inférieure à la limite physique définie dans le fichier de paramètres (**vaplocat.ini**), il est envoyé en une seule fois. Sinon, il est découpé en segments de message. La partie Client envoie le premier segment et attend que le Moniteur de Communication VA Pac lui retourne un accusé de réception pour émettre le suivant, et ainsi de suite jusqu'au dernier segment du message.

Le Moniteur de Communication reçoit chaque segment de message dans la Queue IMS via l'instruction **GU**. Après réception du dernier segment, il reconstitue le message logique complet puis traite la requête. De la même manière que pour l'émission du message de requête, si la longueur du message de réponse est inférieure à la longueur maximale définie, la réponse est retournée au Client en une seule fois avec l'instruction **ISRT**. Sinon, le Moniteur applique le même principe de découpage et d'envoi de segments que pour le message de requête.

Le ***Format*** de message MQSeries utilisé est du type **MQFMT_IMS**:

<MQIIH><LLZZ><Transcode><Application data>

LL

Longueur du segment qui suit (longueur maximale d'un segment IMS=32764 octets)

ZZ=00

Transcode

Code transaction IMS (8 caractères)

MQIIH

En-tête du Bridge IMS :

- **Format = MQFMT_IMS_VAR_STRING**
- **ReplyToFormat = MQFMT_IMS_VAR_STRING**
- **Authenticator** = mot de passe de l'utilisateur indiqué dans **MQMD**
- **CommitMode = 0 (commit then send)**
- **SecurityScope = MQISS_FULL**

  Contrôle de sécurité au niveau Regions de contrôle et dépendante

Définition des objets MQSeries

Les objets MQSeries sont documentés au point *Définition des objets MQSeries* pour le protocole MQSeries , mais ils comportent les spécificités suivantes pour le protocole MQSeries – IMS Bridge :

- **Bridge Storage Class**

  Définition d'une **Storage Class** en spécifiant les noms du group et du membre XCF
- **Request Queue**

  Queue locale du serveur définie avec la **Storage Class** du Bridge.

Configuration

La configuration du protocole de communication passe par la déclaration de paramètres dans le fichier des localisations **vaplocat.ini**. (pour des explications sur ce fichier, voir le *Chapitre 5 : Notion de localisation*).

Les paramètres de la partie cliente de ce protocole de communication sont documentés à l'*Annexe : Paramétrage des logiciels externes*.

Paramètres obligatoires

COMM_TYPE=MQIMS ou
COMM_TYPE=MQMIMS si le middleware est installé sur la même machine que MQSeries Server et que l'on souhaite utiliser un lien direct avec MQSeries.

MONITOR

Nom du programme Moniteur de Communication VA Pac.

Ce Moniteur de Communication doit être généré avec le type **MQBRIDGE**.

MESSAGE_LENGTH

Longueur maximum du message attendu par le Moniteur de Communication.

IXO_QUEUEMANAGER
Nom du Queue Manager local de l'application Cliente

IXO_REQUESTQUEUE

Nom de la Queue des messages de requête

IXO_REPLYQUEUE

Nom de la Queue des messages de réponse

IXO_TRANSID

Nom de la transaction du Moniteur de Communication

Paramètres facultatifs

IXO_TIMEOUT

Temps d'attente de réception d'une réponse, exprimé en secondes (30 secondes par défaut).

IXO_LOCALCCSID

Code page local (819 par défaut), utilisé pour la conversion des messages par MQSeries lors des lectures et écritures des Queues.

IXO_HEADERVERSION

Version de la structure **MQIIH** : 1(défaut) ou 2, en fonction de l'interface IMS-Bridge utilisée (1 car. max.)

IXO_DYNAMICREPLYQUEUE

Lorsque ce paramètre est renseigné, la Queue de réponses est créée dynamiquement par MQSeries pour pouvoir transmettre les messages de réponse (cf. Documentation MQSeries pour l'utilisation de ce type de Queue).

IXO_REQUESTEXPIRY

Délai d'expiration d'une requête, exprimé en secondes.
C'est la durée au delà de laquelle une requête (réponse incluse) est considérée comme perdue. Tous les messages relatifs à cette requête seront alors automatiquement purgés par le gestionnaire MQSeries.
Infini par défaut.

Ne pas définir **HOST_ENCODING**, car le transcodage est pris en charge par MQSeries.

Pour l'architecture distribuée DQM, les Channels émetteurs doivent avoir l'option **CONVERT** positionnée à **YES**.

EXCI

Prérequis

Ce type de middleware ne fonctionne que sur la plateforme OS390 V2R9 avec CICS Transaction Server 1.3.

Description du fonctionnement

Le middleware utilise l'API **EXCI Call Interface** dans son implémentation.

Il établit une connexion en appelant successivement les fonctions :

- **Initialise_Use**,
- **Allocate_Pipe**,
- **Open_Pipe**.

Les messages sont ensuite échangés via la fonction :

- **DPL_Request**

Lorsque le dialogue avec le serveur est terminé, la connexion est refermée par appel aux fonctions:

- **Close_Pipe**,
- **Deallocate_Pipe**.

Configuration

La configuration du protocole de communication passe par la déclaration de paramètres dans le fichier des localisations **vaplocat.ini**. (pour des explications sur ce fichier, voir le *Chapitre 5 : Notion de localisation*).

Paramètres obligatoires

COMM_TYPE=EXCI

MONITOR

Nom du programme Moniteur de Communication VA Pac.

↳ Ce Moniteur de Communication doit être généré avec le type **CICS**.

MESSAGE_LENGTH

Longueur maximum du message attendu par le Moniteur de Communication.

IXO_NETNAME

Identifiant de l'utilisateur EXCI ou attribut NETNAME affecté à la CONNECTION définie pour le Pipe (8 car. max).

IXO_CICSAPPLID

Applid de la région CICS (8 car. max).

IXO_TRANSID

Nom de la transaction du Moniteur de Communication (4 car. max.).

Socket

Prérequis

TCP/IP

Description du fonctionnement

Le protocole adopté par le middleware repose sur l'utilisation du listener Socket VisualAge Pacbase.

Le listener est détaillé dans le *Chapitre 8 : Utilisation du listener VA Pac*.

Configuration

La configuration du protocole de communication passe par la déclaration de paramètres dans le fichier des localisations **vaplocat.ini**. (pour des explications sur ce fichier, voir le *Chapitre 5 : Notion de localisation*).

Les paramètres de la partie cliente de ce protocole de communication sont documentés à l'*Annexe : Paramétrage des logiciels externes*.

Paramètres obligatoires

COMM_TYPE=SOCKET

MONITOR

Nom du programme Moniteur de Communication VA Pac.

Ce Moniteur de Communication doit être généré avec le type **SOCKET**.

MESSAGE_LENGTH

Longueur maximum du message attendu par le Moniteur de Communication.

IXO_ADDRESS

Ce paramètre (30 caractères maximum) doit avoir le format :

0x0002ppppaaaaaaaa, avec :

0x0002 : mode d'adressage standard Internet (AF_INET)

pppp : numéro de port en hexadécimal

aaaaaaaa : adresse IP en hexadécimal

ou **Host Port**, avec :

Host: adresse IP au format **a.b.c.d** ou nom logique du serveur

Port: numéro de port en décimal, 5 caractères maximum

Paramètre facultatif

IXO_LISTENERCHARCODE

Valeur du jeu de caractères codés. Ce paramètre doit être positionné à **EBCDIC** lorsque le listener VisualAge Pacbase est sur une plateforme AS400.

IXO_TIMEOUT

Temps d'attente de réception d'une réponse, exprimé en secondes (30 secondes par défaut).

Paramètres de génération du Moniteur de Communication

COMM_TYPE=SOCKET

Socket CICS

Prérequis

CICS TCP/IP Sockets Interface V3.1

Description du fonctionnement

Le protocole adopté par le middleware repose sur l'utilisation du listener Socket CICS standard: transaction **CSKL** appelant le programme **EZACIC02**.

- Le paramétrage du lancement de ce listener s'effectue à l'aide de la transaction **CSKE**. On spécifie entre autres le numéro de port sur lequel le listener doit se mettre à l 'écoute.
- Après la connexion d'un client sur ce port, celui-ci doit transmettre un premier message au listener. Ce message initiateur doit avoir une des structures requises par le listener (**Listener Input Format**).

  La structure retenue est : **CodeTransaction,MessageAppli**
  avec
  - **CodeTransaction**

    transaction CICS qui lance le Moniteur de Communication VA Pac (4 car.)
  - **,**

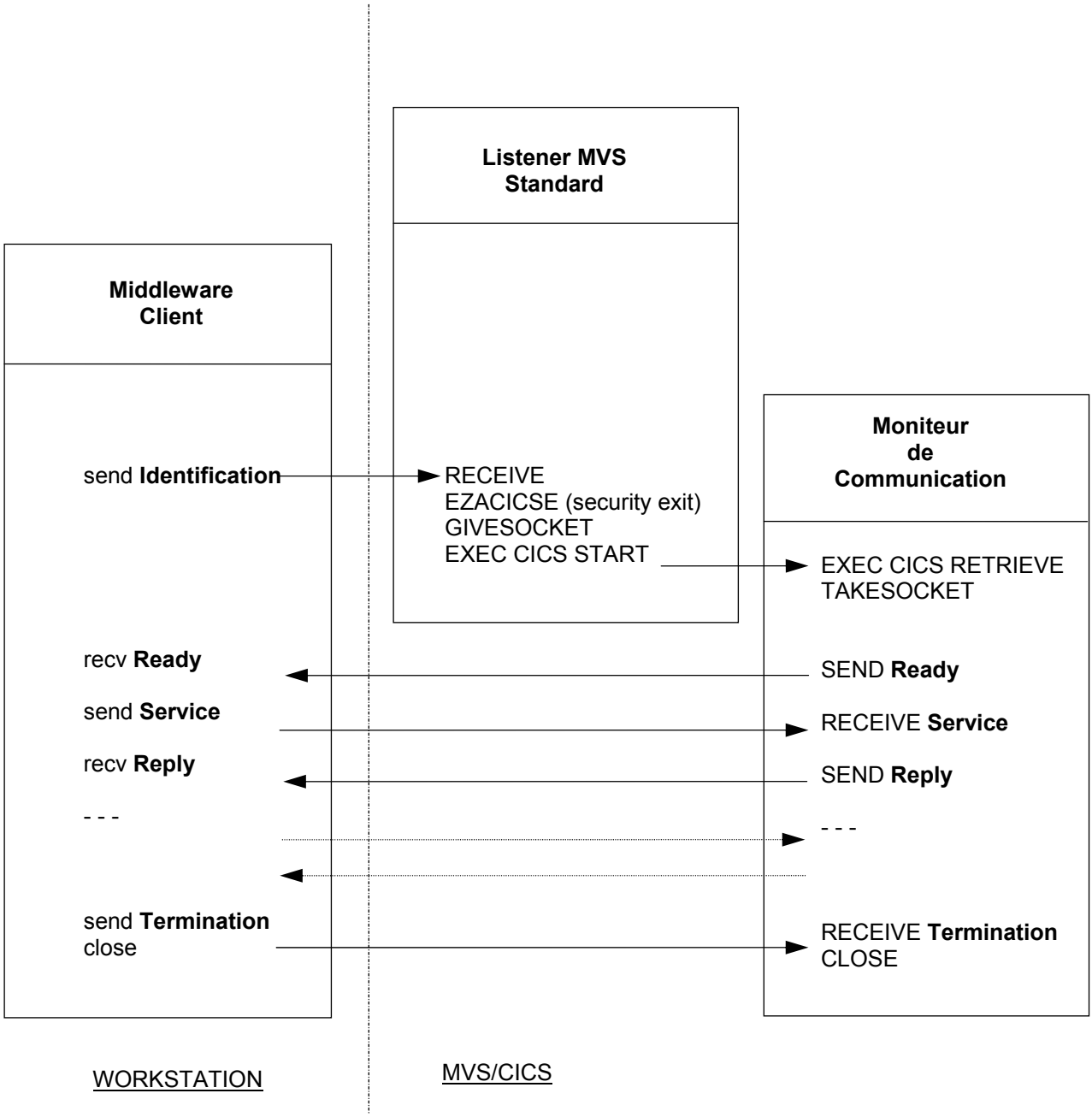
    séparateur (1 car)
  - **MessageAppli**

    message transmis au Moniteur de Communication VA Pac (35 car. max.)
- Après réception de ce message, le listener lance la transaction serveur en lui transmettant une zone Cobol ayant la structure suivante (**Listener Output Format**) :

```
01          TCPSOCKET-PARM.
   05       GIVE-TAKE-SOCKET    PIC 9(8) COMP.
   05       LSTN-NAME           PIC X(8).
   05       LSTN-SUBNAME        PIC X(8).
   05       CLIENT-IN-DATA      PIC X(35).
   05       FILLER              PIC X(1).
   05       SOCKADDR-IN-PARM.
      10    SIN-FAMILY          PIC 9(4) COMP.
      10    SIN-PORT            PIC 9(4) COMP.
      10    SIN-ADDRESS         PIC 9(8) COMP.
      10    SIN-ZERO            PIC X(8).
```

  Le message applicatif est transmis via la zone **CLIENT-IN-DATA**.
- A partir de cet instant, la transaction Serveur est en communication avec l'application Cliente connectée.

Le module de contrôle de sécurité n'est pas fourni en standard, mais un point de sortie est prévu pour permettre l'implémentation de cette fonction. Ce module doit être développé comme étant un programme CICS dont le nom est obligatoirement **EZACICSE**. Il est exécuté systématiquement par le listener, avant le déclenchement de la transaction serveur, par un **EXEC CICS LINK** et en transmettant les informations de sécurité via la **COMMAREA** (cf. CICS TCP/IP Socket Interface Guide and Reference V3R1).



Configuration

La configuration du protocole de communication passe par la déclaration de paramètres dans le fichier des localisations **vaplocat.ini**. (pour des explications sur ce fichier, voir le *Chapitre 5 : Notion de localisation*).

Les paramètres de la partie cliente de ce protocole de communication sont documentés à l'*Annexe : Paramétrage des logiciels externes*.

Paramètres obligatoires

COMM_TYPE=TCPMVS

MONITOR

Nom du programme Moniteur de Communication VA Pac.

Ce Moniteur de Communication doit être généré avec le type **SOCKET**.

MESSAGE_LENGTH

Longueur maximum du message attendu par le Moniteur de Communication.

IXO_TRANSID

Nom de la transaction du Moniteur de Communication (4 car.)

IXO_ADDRESS

Ce paramètre (30 caractères maximum) doit avoir le format :

0x0002ppppaaaaaaaa, avec :

0x0002 : mode d'adressage standard Internet (AF_INET)

pppp : numéro de port en hexadécimal

aaaaaaaa : adresse IP en hexadécimal

ou **Host Port**, avec :

Host: adresse IP au format **a.b.c.d** ou nom logique du serveur

Port: numéro de port en décimal, 5 caractères maximum

Paramètre facultatif

HOST_ENCODING

Valeur du code page serveur. Cette valeur doit être définie dans le fichier des tables de conversion code page (**charconv.txt**) fourni dans le package middleware.

IXO_TIMEOUT

Temps d'attente de réception d'une réponse, exprimé en secondes (30 secondes par défaut).

Paramètres de génération du Moniteur de Communication

COMM_TYPE=SOCKET

WAITINT

Temps d'attente (en secondes) avant l'arrêt automatique du Moniteur de Communication
30 minutes par défaut.

TDS-TCP/IP

Architecture

```
                    Windows ou Aix    |  GCOS 7
                                      |
Composants Client <-> API TDS-TCP/IP <-> OPEN7/SOCKG7 <-> TDS <-> Composants
Serveur
                                      |
```

Prérequis

Les produits Bull requis sont les suivants:

- sur GCOS7
  - GCOS 7-V7 (minimum TS7560) ou GCOS 7-V8 (minimum TS8560) ou GCOS 7-V9 (minimum TS9662)
  - TDS-TCP/IP
  - SOCKG7 V4.1.0
  - GCOS 7 OPEN 7 release V5
- Client Windows
  - TDS-TCP/IP API for Workstation
- Client AIX (4.3)
  - TDS-TCP/IP API for Unix

Votre application TDS doit être générée avec l'option TCP-IP.

Cinématique des échanges

```
- Client Windows/Aix -                  - GCOS7 TDS TCPIP -
-------------------------------------------------------------------------------

tpconnect                         -->   Connection request, LOGON

tprecv "READY"                    <--   SEND "READY"

tpsend "<MONITOR> <requêtePCV>"   -->   Lance <MONITOR> avec requête dans COMMAREA
         8 car.

tprecv "réponse" ou "erreur"      <--   RECEIVE,
                                        SEND "IXO&_____<réponsePCV>" ou "erreur"
                                                5 spaces
tpsend "BYE"                      -->   LOGOUT

tprecv ""                         <--   SEND ""

tpdiscon
```

Configuration

La configuration du protocole de communication passe par la déclaration de paramètres dans le fichier des localisations **vaplocat.ini**. (pour des explications sur ce fichier, voir le *Chapitre 5 : Notion de localisation*).

Les paramètres de la partie cliente de ce protocole de communication sont documentés à l'*Annexe : Paramétrage des logiciels externes*.

Paramètres obligatoires

COMM_TYPE=TCPTDS

MONITOR

Nom du programme Moniteur de Communication VA Pac.

Ce Moniteur de Communication doit être généré avec le type **TCPTDS**.

MESSAGE_LENGTH

Longueur maximum du message attendu par le Moniteur de Communication (valeur : 32000 max.).
Il faut veiller à ce que la longueur maximum définie par la clause **MESSAGE-LENGTH** dans la **TDS SECTION** de **TDSGEN** soit supérieure à la longueur du message défini ici, plus les 9 caractères de l'en-tête **IXO**, afin d'éviter une erreur de protocole.

IXO_TRANSID

Nom de la transaction du Moniteur de Communication (8 car.).

IXO_HOSTNAME

Nom du host DPS 7000. L'adresse IP n'est pas acceptée (15 car. max.).

IXO_TDSNAME

Nom du TDS. (4 car. max.).

Paramètres facultatifs

HOST_ENCODING

Valeur du code page serveur. Cette valeur doit être définie dans le fichier des tables de conversion code page (**charconv.txt**) fourni dans le package middleware.

IXO_PROJECT

Nom du projet. Une fois connectée, l'application client peut lancer toutes les transactions autorisées pour ce projet en fonction du code PROJECT/TDS indiqué dans le catalogue GCOS 7. Si ce paramètre ne comporte que des espaces, le projet GCOS 7 par défaut est pris en compte. (12 car. max.).

IXO_BILLING

Le compte est vérifié dans le catalogue GCOS 7. Si le paramètre ne comporte que des espaces, le compte GCOS 7 par défaut alloué au projet de cet utilisateur sera retenu. (12 car. max.).
IXO_DATACONVERT

Conversion automatique des données, ASCII/EBCDIC.

Si cette option est activée (="**Y**", par défaut), la conversion ASCII/EBCDIC du buffer data est prise en charge automatiquement par l'interface ATMI.

La conversion des données applicatives peut être également prise en charge par l'application; exemple: les Adapters middleware, en positionnant le paramètre **HOST_ENCODING=297**.

Le message finalement envoyé par **tpsend** est composé du code transaction TDS suivi d'un espace puis de la donnée applicative. Par conséquent, si on n'active pas la conversion automatique, la couche IXO devra prendre en charge la conversion de la partie en-tête du message: le code transID et le caractère Espace.

Les options de conversion **IXO_DATACONVERT** et **HOST_ENCODING** doivent être exclusives afin d'éviter la double conversion.

CPI-C

Prérequis

IBM Personal Communication 4.3

Description du fonctionnement

Ce type de communication est implémenté en utilisant l'API CPI-C (Common Programming Interface - Communications) standard X/Open.

Chaque échange avec le moniteur de communication VA Pac se décompose côté client en appels successifs de fonctions suivantes :

- initialisation de la conversation (cminit)
- positionnement des paramètres de conversation (cmscst, cmscsu..)
- connexion (cmallc)
- envoie de la requête (cmsend)
- réception de la réponse (cmrcv)
- fermeture de connexion (cmdeal)

Configuration

La configuration du protocole de communication passe par la déclaration de paramètres dans le fichier des localisations **vaplocat.ini**. (pour des explications sur ce fichier, voir le *Chapitre 5 : Notion de localisation*).

Les paramètres de la partie cliente de ce protocole de communication sont documentés à l'*Annexe : Paramétrage des logiciels externes*.

Paramètres obligatoires

COMM_TYPE=CPIC

MONITOR

Le "Symbolic Destination Name" défini dans le fichier de configuration CPI-C et associé à la transaction du Moniteur de Communication VA Pac.

Par convention, ce "Symbolic Destination Name" correspond au nom du Moniteur de Communication VA Pac.

Ce moniteur de communication doit être également généré avec le type CPIC.

MESSAGE_LENGTH

Longueur maximum du message attendu par le Moniteur de Communication.

Paramètres facultatifs

HOST_ENCODING

Valeur du code page serveur. Cette valeur doit être définie dans le fichier des tables de conversion code page (**charconv.txt**) fourni dans le package middleware.

IXO_TIMEOUT

Temps d'attente de réception d'une réponse, exprimé en secondes (30 secondes par défaut).

TUXEDO

Prérequis

Tuxedo 8.1

Description du fonctionnement

Ce type de communication est implémenté en utilisant l'API ATMI.

Chaque appel au Moniteur de Communication consiste en un appel **tpcall** avec pour nom de service le nom du moniteur, l'ouverture d'une connexion vers le serveur ayant été préalablement effectuée via un appel à **tpinit**.

Configuration

La configuration du protocole de communication passe par la déclaration de paramètres dans le fichier des localisations **vaplocat.ini**. (pour des explications sur ce fichier, voir le *Chapitre 5 : Notion de localisation*).

Les paramètres de la partie cliente de ce protocole de communication sont documentés à l'*Annexe : Paramétrage des logiciels externes*.

Paramètres obligatoires

COMM_TYPE=TUXEDO pour TUXEDO/WS Mono-Thread
ou
COMM_TYPE=TUXEDOMT pour TUXEDO/WS Multi-Thread

MONITOR

Nom du Moniteur de Communication défini comme service TUXEDO.

IXO_ADDRESS

Ce paramètre (30 caractères maximum) doit avoir le format :

0x0002ppppaaaaaaaa, avec :
0x0002 : mode d'adressage standard Internet (AF_INET)
pppp : numéro de port en hexadécimal
aaaaaaaa : adresse IP en hexadécimal

ou **Host Port**, avec :
Host: adresse IP au format **a.b.c.d** ou nom logique du serveur
Port: numéro de port en décimal, 5 caractères maximum

Paramètres facultatifs

IXO_PASSWORD

Mot de passe de l'utilisateur ("user authentication password"). 30 caractères maximum.
Ce type de mot de passe est administré par le système de sécurité (Standard Tuxedo, Kerberos, …) qui fournit le service d'authentification utilisé et qui est spécifié dans le fichier **UBBCONFIG** de l'application.
Ce paramètre est obligatoire si l'option de sécurité **USER_AUTH** est définie au niveau du serveur TUXEDO (dans **UBBCONFIG**)
Il est préférable de transmettre ce paramètre par l'application cliente.

IXO_APPLPASSWORD

Mot de passe de l'application TUXEDO ("application password"). 30 caractères maximum.
Ce mot de passe est généralement fixé au départ et ne devrait plus évoluer. Il est défini à chaque génération du fichier de configuration de TUXEDO (**UBBCONFIG**).
Il est obligatoire si l'option de sécurité **USER_AUTH** est définie au niveau du serveur TUXEDO (dans **UBBCONFIG**).

IXO_CLTNAME

Nom du client géré au niveau de l'application serveur TUXEDO. (30 caractères maximum)
Le couple **User Name / Client Name** identifie un client de l'Application TUXEDO. Il est utilisé pour contrôler le droit d'accès à l'Application et non pour authentifier l'utilisateur.
Sa présence dépend de la configuration de la sécurité côté Application Serveur TUXEDO.

IXO_GRPNAME

Le **Group Name** est utilisé pour associer un client de l'application à un Resource Manager. (30 caractères maximum.)
Sa présence dépend de la configuration de la sécurité côté Application Serveur TUXEDO.

IXO_FMLCONVERSIONFILE

Conversion en FML
Nom du fichier au format FML utilisé pour le formattage du buffer d'entrée.

NOTE : vous devez également indiquer le chemin d'accès dans la variable d'environnement **FLDTBLDIR** et le nom du fichier dans la variable d'environnement **FIELDTBLS**

HOST_ENCODING

Valeur du code page serveur. Cette valeur doit être définie dans le fichier des tables de conversion code page (**charconv.txt**) fourni dans le package middleware.

IXO_TIMEOUT

Temps d'attente de réception d'une réponse, exprimé en secondes (30 secondes par défaut).

Utilisation de l'API Jolt de BEA

Au lieu d'utiliser l'API TUXEDO classique (impliquant l'utilisation des fichiers **ixotux** et **ixotuxmt**), vous pouvez utiliser l'API Jolt de BEA. Cette API, écrite en Java, est accessible via **JoltAdapter**.

Lancement du repository

Jolt utilise un 'repository' (référentiel) dans lequel sont décrits les services et les buffers utilisés. Vous devez lancer le repository et le listener Jolt en ajoutant les lignes suivantes dans le fichier de configuration TUXEDO :

```
JREPSVR SRVGRP=GROUP2 SRVID=6
  CLOPT="-A -- -W -P /home/ptpc/TuxServ/repository/jrepository"

JSL     SRVGRP=GROUP2 SRVID=5
  CLOPT="-A -- -n //brutus:55555 -m 1 -M 30 -x 10 -T 60"
```

Dans le classpath de l'application voulant utiliser Jolt, vous devez indiquer les chemins pour atteindre **JoltRun.jar**, qui se trouve dans le répertoire **ebusinesstolls\lib**, et les **.jar** de BEA (**jolt.jar**, **joltjse.jar**).. Dans ce cas, vous devez positionner la **PrimaryServerAddress** de JoltAdapter à :

```
//brutus:55555
```

Déclaration des services et buffers

Les buffers utilisés pour la communication sont de type **C_ARRAY** et se nomment **VAPINPUT** et **VAPOUPUT**.

Un utilitaire permet de déclarer les services et les buffers en utilisant un fichier texte. Il est aussi possible de faire la même chose grâce à un outil graphique (référez-vous à la documentation sur l'éditeur BEA Jolt, disponible à l'adresse http://edocs.beasys.com/tuxedo/tux81/jdg/dvrepos.htm)

Cet utilitaire se lance par le script suivant :

```
java -classpath $CLASSPATH/jolt.jar:$CLASSPATH/joltadmin.jar
bea.jolt.admin.jbld -u jpr //brutus:55555 BulkLoadFile
```

avec `BulkLoadFile` contenant :

```
service=MCTXFI
export=true
inbuf=CARRAY
outbuf=CARRAY
param=VAPINPUT
type=carray
access=in
param=VAPOUTPUT
type=carray
access=out
```

GTEA – ECI

Pour toute information sur ces protocoles, veuillez vous adresser au Support Technique.

Chapitre 10 : Résolution des problèmes de communication

Erreurs de communication

Les erreurs dans la chaîne de communication avec le serveur provoquent l'erreur `com.ibm.vap.generic.CommunicationError`. Comme toute erreur Java, une erreur de communication VA Pac ne renvoie pas de clé. Pour identifier la cause de l'erreur, il faut récupérer le message associé à l'erreur (méthode `getMessage()` de la classe).

Trois messages d'erreur de communication sont susceptibles de s'afficher :

- `Erreur ouverture serveur`
- `Erreur appel serveur`
- `Erreur fermeture serveur`

Si un message d'erreur de communication apparaît, vous devez d'abord en informer le responsable de la communication car la cause peut être un encombrement de ligne, une ligne défectueuse, un serveur indisponible...

Vous pouvez aussi trouver l'origine du problème en consultant la trace.

Utilisation de la trace

La trace vous permet d'analyser les problèmes de communication rencontrés.

Trois traces peuvent être exploitées :

- gateway, si celle-ci est utilisée. Dans ce cas, le nom du fichier trace doit être indiqué sur la ligne de lancement de la gateway,
- adaptateur de communication (classe `MiddlewareAdapter`),
- interface de communication (DLLs `IXO`).

Trace gateway

Vous indiquez le fichier de trace gateway via l'option `-tf` et le niveau de trace via l'option `-t`. Les niveaux de trace possibles sont :

- `0` : aucune trace
- `1` : traces d'erreurs (défaut)
- `3` : traces générales
- `5` : traces détaillées

Voici un exemple de trace obtenu lors du lancement de la gateway en interactif, avec un niveau de trace égal à `1` :

```
[VapGateway 17:51:28:94 Parameters:
 LocationsFile=C:\TstMware\tools\vaplocat.ini,
CodePageFile=N:\MwTeam\TestMware\tools\CharConv.txt]
[VapGateway 17:51:28:94 Traces:
 File=C:\TstMware\VapTrace\VapGateway_020329_1751
 28_902.log, Level=1]
[VapGateway 17:51:28:94 Client connections: Timeout=30s]
[VapGateway 17:51:28:94 Server connections: Min=0, Max=No,
Cleaning=60s, ConnectionTimeout=Infinite]
```

```
[VapGateway 17:51:28:94 Address: Host=pc5548hd
 (9.101.40.17), Port=50000]
[VapGateway 17:51:28:94 Waiting for client connection]
```

Trace MiddlewareAdapter

MiddlewareAdapter est une classe fournie dans le runtime VA Pac (**vaprun.jar**) et est utilisée dans le cas d'une communication en mode direct vers le serveur. Elle fournit une API permettant de définir le contexte de communication vers l'application serveur, d'envoyer des requêtes de service et de recevoir les messages de réponse. Comme pour **VapGateway**, il est possible de tracer l'ensemble de ces traitements en faisant appel aux méthodes suivantes de l'objet **MiddlewareAdapter** associé aux proxies utilisées par l'application :

- **setTraceLevel (Int TraceLevel)**,
- **setTraceFile (String TraceFile)**.

Ces méthodes sont documentées dans le Manuel "Interface publique des composants générés".

Trace IXO

Deux versions de DLL de communication sont livrées : la version Strip et la version Trace.

La version Strip correspond à la version optimisée des DLLs du middleware. Ces DLLs ne contiennent pas le code permettant de tracer le fonctionnement des communications et n'interprètent donc pas les variables d'environnement réservées à cet effet.

A l'installation, les DLLs prêtes à être exécutées sont celles de la version Strip.

Les fichiers DLL de la version Trace sont livrés avec l'extension **.001**.
Pour activer la DLL version Trace, vous devez donc changer l'extension **.001** de la DLL **IXO** qui correspond au protocole de communication utilisé en **.DLL**, **.SO** ou **.O**, selon la plateforme d'exécution.

*Par exemple, si vous utilisez le protocole MQSeries sous Windows, vous devez renommer le fichier **IXOMQS.001** en **IXOMQS.DLL**.*

Quand la DLL trace est en service, vous activez l'écriture des traces à partir des deux variables d'environnement suivantes :

- **IXOTRACE** :
  Cette variable permet d'activer (**IXOTRACE=1**) ou de désactiver (**IXOTRACE=0**) la trace de l'API Middleware.
- **IXOTRACE_FILE** :
  Cette variable permet de spécifier, quand la trace est active (**IXOTRACE=1**), le chemin du fichier de trace.
  exemple : **IXOTRACE_FILE=c:\tmp\ixo_err.txt**

Le fichier des traces **IXO** n'est jamais réinitialisé par les fonctions middleware. Pour éviter un effet cumulatif indésirable, il est donc conseillé de détruire ce fichier systématiquement ; sachant qu'il sera recréé automatiquement.

N'oubliez surtout pas de remettre la version Strip en service quand vous n'aurez plus besoin de trace.

Chapitre 11 : Déploiement du middleware

Le middleware est livré sous forme d'un fichier compressé par plateforme cible dans le répertoire **middleware**, accompagné d'un fichier **readme** que vous devez lire avant de procéder à l'installation manuelle des composants middleware. Il est conseillé d'installer entièrement le package correspondant à la plateforme où va s'exécuter le middleware.

Pour installer, vous devez :

- transférer le package vers la machine cible,
- décompresser et extraire les fichiers en les plaçant dans le répertoire prévu pour le middleware VA Pac, en utilisant WinZip ou la commande **tar** d'Unix.
  Exemple de commande d'extraction sur Solaris :
  >zcat vm300v06_solaris.tar.z | tar -xvf -

  Pour les systèmes Unix, vous devez changer les attributs **owner** et **group** de tous les fichiers extraits. Par exemple :
  >chown monident *
  >chgrp mongroupe *

Annexe : Paramétrage des logiciels externes

Un paramétrage spécifique des logiciels externes est nécessaire à la mise en œuvre des communications pour une application générée par les modules eBusiness et Pacbase WebConnection.

Les fiches de configuration qui suivent correspondent à des configurations testées. Elles sont donc dépendantes d'un environnement technique particulier. Par exemple pour les communications entre les plateformes Windows et le host MVS, les postes de travail du réseau Token-Ring s'adressent à une gateway SNA Communication Manager ou SNA Server, elles-mêmes s'adressant à un contrôleur IBM 3745 pour atteindre le host MVS.

Elles ne proposent donc qu'une solution de paramétrage pour un contexte particulier et doivent être adaptées aux impératifs techniques de chaque site.

IMS CPI-C

Configuration MVS et IMS

Définitions VTAM

Minima requis : VTAM Version 3.3

Définition de l'ATCSTR de VTAM

```
*******************************
NOPROMPT,CONFIG=00,SSCPID=01,
MAXSUBA=31,SUPP=NOSUP,
SSCPNAME=A01M,
SSCPORD=DEFINED,
NETID=NETCGI,
HOSTSA=1,
CRPLBUF=(550,,20,,40,40),
IOBUF=(420,182,25,,40,40),
LFBUF=(300,,0,,20,10),
LPBUF=(50,,0,,5,5),
SFBUF=(50,,0,,5,5),
SPBUF=(90,,0,,5,5),
NOTRACE,TYPE=VTAM
*******************************
```

MAC Address du TIC (attachement LAN) contrôleur 3745 dans NCP :

```
*********************************************
*            TIC BNN
*********************************************
A01L1TK  LINE ADDRESS=(1088,FULL),
                       PORTADD=01,
                       LOCADD=400003172000,
                       ISTATUS=ACTIVE,
                              UACB=(X$P1AX,X$P1AR)
*
A01TK1PU PU ISTATUS=ACTIVE,
                       ADDR=01
*********************************************
```

Définition APPC/IMS

Une application spécifique APPC/IMS pour LU6.2 doit être définie dans VTAM (différente de l'APPL IMS utilisée pour les terminaux 3270).

```
*/* LIB: SYS1.VTAMLST(A0100)
*/*
A01IMS62 APPL    EAS=50,                ESTIMATED CONCURRENT SESSIONS
*
         MODETAB=MTLU62,                      nom de la table des modes
         DLOGMOD=LU62,                        nom du mode dans la table
         APPC=YES,                            paramètre obligatoire
         ACBNAME=A01IMS62,       APPLID FOR ACB
         AUTH=(ACQ,BLOCK,PASS)   IMS CAN ACQUIRE & PASS TMLS
```

Définition du mode

- Définition des caractéristiques pour les Sessions LU6.2
- Le Mode SNASVCMG est utilisé avec le support "Parallel Sessions".

```
TITLE '--- "MODTABLE" CONCERNANT LES LU 6.2 ---'
*
MTLU62   MODETAB
         SPACE 4
SNASVCMG MODEENT LOGMODE=SNASVCMG, FMPROF=X'13',
               TSPROF=X'07', PRIPROT=X'B0', SECPROT=X'B0',
               COMPROT=X'D0B1', RUSIZES=X'8585', ENCR=B'0000',
               PSERVIC=X'060200000000000000000300'
LU62     MODEENT LOGMODE=LU62, TYPE=X'00', FMPROF=X'13',
               TSPROF=X'07', PRIPROT=X'B0', SECPROT=X'B0',
               COMPROT=X'50B1', RUSIZES=X'8787', SRCVPAC=X'00',
               PSNDPAC=X'00', SSNDPAC=X'00',
               PSERVIC=X'06020000000000000002C00'
```

Définition de la passerelle SNA

```
*/* LIB: SYS1.VTAMLST(SW1TKR)
*/*
*/*
*/* SWITCHED MAJOR NODE TOKEN-RING ST-MARC :            -  06/07/95.
*/*   --->    LIEN XCA MAJOR NODE ==> XCA1TKR  (IBM3172-3)
*/*   --->    LIEN GROUPE XCA     ==> GRP02
*            ------------------------------
*    - MODEL FOR IDBLK X'05D' - OS/2 COMMUNICATIONS MANAGER
*            ------------------------------
*
SW1TKR   VBUILD TYPE=SWNET,MAXNO=99,MAXGRP=10
*
*          ----------------------------------------
* -------> DEFINING A GATEWAY TOKEN-RING --> GTWTK1  <--------
*          ----------------------------------------
W1TK00   PU    ADDR=50,
               CPNAME=GTWTK1,
               IDBLK=05D,
               IDNUM=00002,
               DYNLU=YES,
               MAXPATH=1,
               DISCNT=NO,
               IRETRY=YES,
               VPACING=7,
               PACING=7,
               SSCPFM=USSSCS,
               MAXDATA=4096,
               PUTYPE=2
```

L'option **DYNLU=YES** permet d'éviter toute définition complémentaire de Lu 6.2 au niveau VTAM (pour les postes du réseau TR).

Définition d'une LU indépendante

Malgré ce qui vient d'être dit ci-dessus, il peut être intéressant de définir une LU indépendante pour les premiers tests de communication :

```
*/* LIB: SYS1.VTAMLST(SW1TKR)
*/*
*/*
*/* SWITCHED MAJOR NODE TOKEN-RING ST-MARC :            -  06/07/95.
*/*
*/*
*  INDEPENDENT LUS
*
IMS4349          LU      LOCADDR=0,
                 ISTATUS=ACTIVE,
                 DLOGMOD=LU62,
                 MODETAB=MTLU62
```

Définitions APPC/MVS

Minima requis : MVS/ESA Version 4.2

Deux membres de la **SYS1.PARMLIB** sont nécessaires pour définir les caractéristiques de la Local LU APPC/MVS et d'ASCH (scheduler APPC).

Définition de la Local LU APPC/MVS

- Paramétrage

```
BROWSE -- SYS1.PARMLIB(APPCPM00) - 01.12 ----------- LINE 0000
 COMMAND ===>
******************************** TOP OF DATA ****************
/*************************************************************/
/*  THE FOLLOWING PARAMETERS ARE FOR THE APPC ADDRESS SPACE. */
/*  THE APPC ADDRESS SPACE HANDLES THE ACTUAL COMMUNICATIONS.*/
/*                                                           */
/*  THESE MEMBERS PROVIDE THE LINKAGE BETWEEN LU NAMES AND   */
/*  TRANSACTION SCHEDULERS.                                  */
/*************************************************************/
LUADD
      ACBNAME(A01IMS62)  □ correspond à l'APPL APPC/IMS de VTAM
      SCHED(CGIB)
      BASE
      TPDATA(UTI.APPCTP)
      TPLEVEL(SYSTEM)
SIDEINFO DATASET(UTI.APPCSI)
```

- Lancement

```
BROWSE -- SYS1.PROCLIB(APPC) ------------------------
 COMMAND ===>
******************************** TOP OF DATA ********
//APPC PROC APPC=00
//APPC EXEC PGM=ATBINITM,PARM='APPC=&APPC',REGION=0K
******************************** BOTTOM OF DATA ******
```

Définition du scheduler APPC/MVS

- Paramétrage

```
BROWSE -- SYS1.PARMLIB(ASCHPM00) - 01.00 -----------
 COMMAND ===>
******************************** TOP OF DATA ********
/*************************************************
/* THE FOLLOWING IS ADDED TO ENABLE APPC/MVS.
/*************************************************
 CLASSADD CLASSNAME(SVSAMP)
      MAX(10)
      MIN(2)
```

```
      RESPGOAL(0.02)
      MSGLIMIT(700)
 OPTIONS DEFAULT(SVSAMP)
      SUBSYS(JES2)
 TPDEFAULT REGION(4M)
      TIME(10,30)
      MSGLEVEL(1,1)
      OUTCLASS(R)
****************************** BOTTOM OF DATA ******
```

- Lancement

```
BROWSE -- SYS1.PROCLIB(ASCH) ----------------------
 COMMAND ===>
******************************** TOP OF DATA ********
//ASCH PROC ASCH=00
//ASCH EXEC PGM=ASBSCHIN,PARM='ASCH=&ASCH',REGION=0K
******************************** BOTTOM OF DATA ******
```

Définitions IMS

Minima requis : IMS Version 4.1

IMSCTRL Macro

IMS doit être généré en utilisant les bibliothèques MVS/ESA correspondant à la version spécifiée en troisième paramètre du mot-clé **SYSTEM**. Cette version MVS/ESA doit être au minimum la 4.2.

```
BROWSE -- EX.IMS410.SOURCE(STAGE1) - 01.28 -------------
 COMMAND ===>
*
* IMSCTRL MACRO --
*
         IMSCTRL  SYSTEM=(VS/2,CTLBLKS,4.2),
               DBRC=(YES,NO),
               DBRCNM=DBRC41,
               DLINM=DLISAS41,
               DCLWA=YES,
               IMSID=CGIB,      correspond au paramètre SCHED de APPCPM00
               NAMECHK=(YES,S1),
               MAXIO=(,015),
               MAXREGN=(008,512K,A,A),
               MCS=(8),
               DESC=7,
               MAXCLAS=020
```

Startup

Pour permettre à IMS d'établir la connexion avec APPC/MVS, il faut spécifier **APPC=Y** dans le Job de Startup IMS (**DFSPBxxx** où **xxx** est le suffixe de la région).

Définition de la transaction

```
BROWSE -- EX.IMS410.SOURCE(STAGE1) - 01.28 ---------
 COMMAND ===>
******** TRANSACTION DB2 POUR BABY CLIENT (PTAB) ********************
APPLCTN  PSB=BABIVG
TRANSACT CODE=BABI,SEGSIZE=00000,MODE=SNGL,SEGNO=00000,
              PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,
              MSGTYPE=(SNGLSEG,RESPONSE,4)
```

IMS Connect

Exemple de fichier de configuration d'IMS Connect

```
IMS.FRIMSCEC.PROCLIB(HWSCFG00)
*********************************************************************************************
HWS (ID=ITOC01,RACF=Y)
TCPIP (HOSTNAME=TCPIP,RACFID=RACF,PORTID=(4000),MAXSOC=300,TIMEOUT=00000,EXIT=(HWSIMSO0))
DATASTORE (ID=IMSC,GROUP=GPACMQ,MEMBER=HWSMEM,TMEMBER=FRIMSCEC)
*********************************************************************************************
```

Exemple de JCL de démarrage d'IMS Connect

```
IMS.FRIMSCEC.JOBS(IMSCTOC)
*********************************************************************************************
//IMSCTOC  JOB (FR9970,FRAIMSC,MM103,NAJT,JC),MSGCLASS=S,CLASS=A
//HWS      EXEC PGM=HWSHWS00,
//            PARM='BPECFG=BPECFG00,HWSCFG=HWSCFG00'
//STEPLIB  DD DSN=IMS.HWS.SHWSRESL,DISP=SHR
//         DD DSN=IMS.FRIMSCEC.RESLIB,DISP=SHR
//PROCLIB  DD DSN=IMS.FRIMSCEC.PROCLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//HWSRCORD DD DSN=IMS.HWS.HWSRCDR,DISP=SHR
*********************************************************************************************
```

CICS CPI-C

Configuration MVS et CICS

Définitions VTAM

Minima requis : VTAM Version 3.3

Définition de l'ATCSTR de VTAM

```
*******************************
NOPROMPT,CONFIG=00,SSCPID=01,
MAXSUBA=31,SUPP=NOSUP,
SSCPNAME=A01M,
SSCPORD=DEFINED,
NETID=NETCGI,
HOSTSA=1,
CRPLBUF=(550,,20,,40,40),
IOBUF=(420,182,25,,40,40),
LFBUF=(300,,0,,20,10),
LPBUF=(50,,0,,5,5),
SFBUF=(50,,0,,5,5),
SPBUF=(90,,0,,5,5),
NOTRACE,TYPE=VTAM
*******************************
```

MAC Address du TIC (attachement LAN) contrôleur 3745 dans NCP :

```
*********************************************
*            TIC BNN                              05742090
*********************************************
A01L1TK  LINE ADDRESS=(1088,FULL),
                      PORTADD=01,
                      LOCADD=400003172000,
                      ISTATUS=ACTIVE,
                      UACB=(X$P1AX,X$P1AR)
*
A01TK1PU PU ISTATUS=ACTIVE,
                      ADDR=01
*********************************************
```

Définition du CICS

```
A01CICS1 APPL  EAS=160,                ESTIMATED CONCURRENT SESSIONS   *
               ACBNAME=CICST,          APPLID FOR ACB                  *
               AUTH=(ACQ,VPACE,PASS),  CICS CAN ACQUIRE & PASS TMLS    *
*                                      CICS CAN REQUEST BLOCKED INPUT

               PARSESS=YES.                    Supports parallel Sessions
               SONSCIP=YES,                                            *
               MODETAB=MTLU62                  nom de la table des modes
```

Définition du mode

- Définition des caractéristiques pour les Sessions LU6.2
- Le Mode SNASVCMG est utilisé avec le support "Parallel Sessions".

```
TITLE '--- "MODTABLE" CONCERNANT LES LU 6.2 ---'
*
MTLU62   MODETAB
         SPACE 4
SNASVCMG MODEENT LOGMODE=SNASVCMG, FMPROF=X'13',
               TSPROF=X'07', PRIPROT=X'B0', SECPROT=X'B0',
               COMPROT=X'D0B1', RUSIZES=X'8585', ENCR=B'0000',
               PSERVIC=X'060200000000000000000300'
LU62     MODEENT LOGMODE=LU62, TYPE=X'00', FMPROF=X'13',
               TSPROF=X'07', PRIPROT=X'B0', SECPROT=X'B0',
               COMPROT=X'50B1', RUSIZES=X'8787', SRCVPAC=X'00',
               PSNDPAC=X'00', SSNDPAC=X'00',
               PSERVIC=X'06020000000000000002C00'
```

Définition de la passerelle SNA

```
*/* LIB: SYS1.VTAMLST(SW1TKR)
*/*
*/*
*/* SWITCHED MAJOR NODE TOKEN-RING ST-MARC :            -  06/07/95.
*/*   --->    LIEN XCA MAJOR NODE ==> XCA1TKR  (IBM3172-3)
*/*   --->    LIEN GROUPE XCA     ==> GRP02
*             -------------------------------
*    - MODEL FOR IDBLK X'05D' - OS/2 COMMUNICATIONS MANAGER
*             -------------------------------
*
SW1TKR   VBUILD TYPE=SWNET,MAXNO=99,MAXGRP=10
*
*          ----------------------------------------
* -------> DEFINING A GATEWAY TOKEN-RING -->  GTWK1  <--------
*          ----------------------------------------
W1TK00   PU    ADDR=50,
               CPNAME=GTWTK1,
               IDBLK=05D,
               IDNUM=00002,
               DYNLU=YES,
               MAXPATH=1,
               DISCNT=NO,
               IRETRY=YES,
               VPACING=7,
               PACING=7,
               SSCPFM=USSSCS,
               MAXDATA=4096,
               PUTYPE=2
```

L'option **DYNLU=YES** permet d'éviter toute définition complémentaire de Lu 6.2 au niveau VTAM (pour les postes du réseau TR).

Définition d'une LU indépendante

Malgré ce qui vient d'être dit ci-dessus, il peut être intéressant de définir une LU indépendante pour les premiers tests de communication :

```
*/* LIB: SYS1.VTAMLST(SW1TKR)
*/*
*/*
*/* SWITCHED MAJOR NODE TOKEN-RING ST-MARC :            -  06/07/95.
*/*
*/*
*  INDEPENDENT LUS
*
CICSFBFB  LU      LOCADDR=0,
                  ISTATUS=ACTIVE,
                  DLOGMOD=LU62,
                  MODETAB=MTLU62
```

Définitions APPC/MVS

Minima requis : MVS/ESA Version 4.2

Il n'y a pas de définitions spécifiques car on utilise la couche APPC livrée avec la version de CICS.

Définitions CICS

Paramètre de l'InterSystem Communication dans la table SIT

ISC=YES

Connexion

```
 Connection     : SGFB
  Group         : GRPISC5
  DEscription   :
CONNECTION IDENTIFIERS
  Netname       : CICSFBFB     à déclarer comme Local LU dans CM/2
                               (codification libre)
  INDsys        :
REMOTE ATTRIBUTES
  REMOTESystem  :
  REMOTEName    :
CONNECTION PROPERTIES
  ACcessmethod  : Vtam
  Protocol      : Appc
  SInglesess    : No           dans le cas d'une LU indépendante
  DAtastream    : User
  RECordformat  : U
OPERATIONAL PROPERTIES
  AUtoconnect   : Yes
  INService     : Yes
SECURITY
  SEcurityname  : PTPD
  ATtachsec     : Verify     Vérification du userid et password à la
                             conversation
  BINDPassword  :
  BINDSecurity  : No
```

Session

```
 Sessions       : SESSIOFB
  Group         : GRPISC5
  DEscription   :
SESSION IDENTIFIERS
  Connection    : SGFB         code de la connexion définie ci-dessus
  SESSName      :
  NETnameq      :
  MOdename      : LU62         mode défini dans la MODTABLE de VTAM
SESSION PROPERTIES
  Protocol      : Appc
  MAximum       : 004 , 002
  RECEIVEPfx    :
  RECEIVECount  :
  SENDPfx       :
  SENDCount     :
  SENDSize      : 08192
  RECEIVESize   : 08192
  SESSPriority  : 000
  Transaction   :
OPERATOR DEFAULTS
  OPERId        :
  OPERPriority  : 000
  OPERRsl       : 0
  OPERSecurity  : 1
PRESET SECURITY
  USERId        :
OPERATIONAL PROPERTIES
  Autoconnect   : Yes
  INservice     :
  Buildchain    : Yes
  USERArealen   : 000
  IOarealen     : 00000 , 00000
  RELreq        : Yes
  DIscreq       : No
  NEPclass      : 000
RECOVERY
  RECOVOption   : Sysdefault
  RECOVNotify   : None
```

Définition de la transaction

```
OBJECT CHARACTERISTICS                                  CICS RELEASE = 0330
   CEDB  View
    TRansaction    : VIC0              code à reporter dans le TP name du
                                       CPIC Side Information de CM/2
    Group          : VISUAL
    DEscription    :
    PROGram        : AVVMSV            code du moniteur de communication
    TWasize        : 00000              0-32767
    PROFile        : DFHCICST
    PArtitionset   :
    STatus         : Enabled            Enabled ! Disabled
    PRIMedsize     : 00000              0-65520
    TASKDATALoc    : Below              Below ! Any
    TASKDATAKey    : User               User ! Cics
   REMOTE ATTRIBUTES
    DYnamic        : No                 No ! Yes
    REMOTESystem   :
    REMOTEName     :
    TRProf         :
    Localq         :                    No ! Yes
   SCHEDULING
    PRIOrity       : 001                0-255
    TClass         : No                 No ! 1-10
   ALIASES
    Alias          :
    TASKReq        :
    XTRanid        :
    TPName         :
                   :
    XTPname        :
                   :
                   :
    RECOVERY
    DTimout        : No                 No ! 1-6800
    Indoubt        : Backout            Backout ! Commit ! Wait
    RESTart        : No                 No ! Yes
    SPurge         : Yes                No ! Yes
    TPUrge         : Yes                No ! Yes
    DUmp           : Yes                Yes ! No
    TRACe          : Yes                Yes ! No
   SECURITY
    RESSec         : No                 No ! Yes
    Cmdsec         : No                 No ! Yes
    Extsec         : No                 No ! Yes
    TRANsec        : 01                 1-64
    RSl            : 00                 0-24 ! Public
```

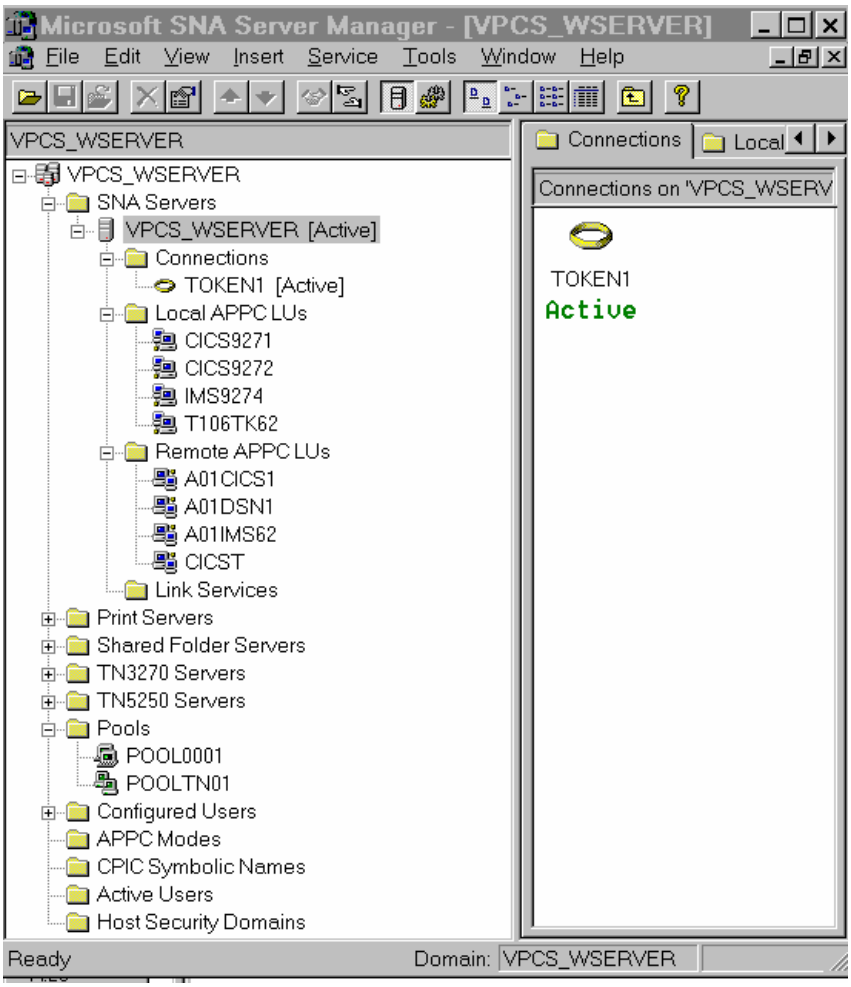
De plus, si la transaction fait des accès à une base DB2, il faut que celle-ci soit autorisée et qu'elle soit rattachée au plan DB2. Le code transaction et le plan DB2 doivent donc être déclarés dans la Table RCT. Dans cet exemple, la transaction VIC0 et le plan DB2 ATDF sont utilisés par l'application Cliente :

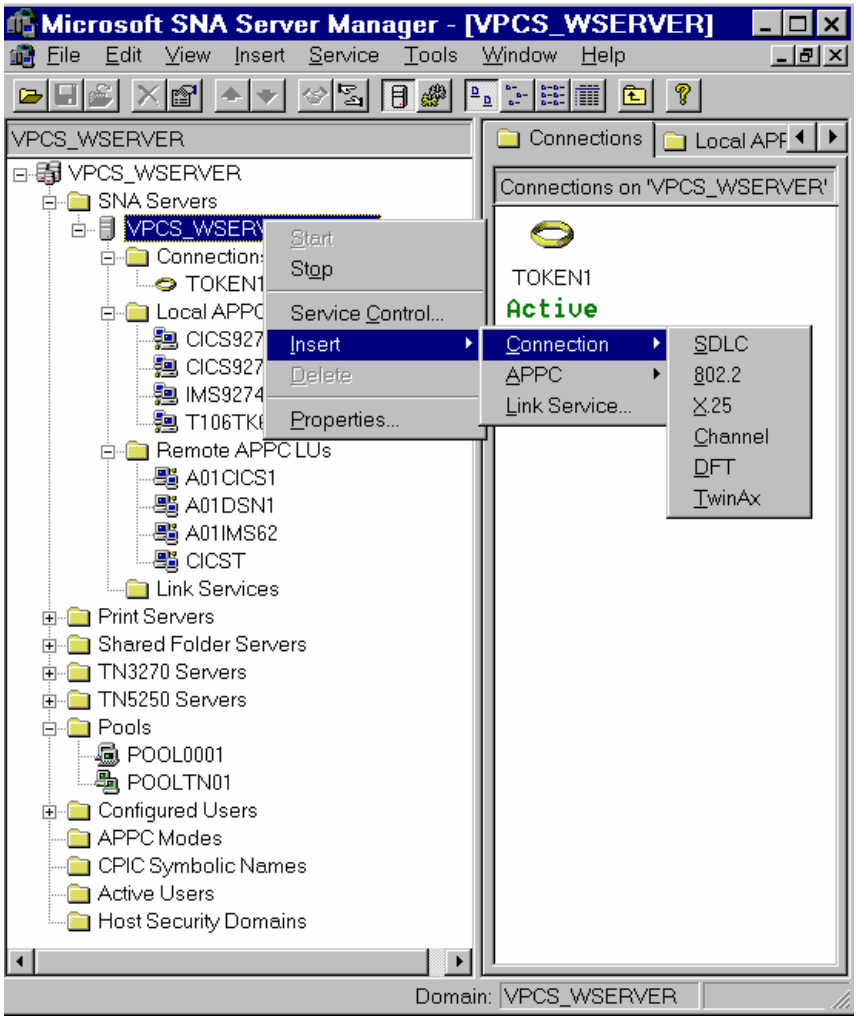
```
DSNRCT TYPE = ENTRY,TXID=(VIC0),
         THRDM=6,THRDA=6,PLAN=ATDF,AUTH=(USERID,*,*)
```

Configuration SNA Server 3.0A pour Windows NT

(Le Service Pack 2 doit être appliqué pour des corrections APPC).

Les paramètres SNA Server doivent correspondre aux définitions VTAM, NCP et CICS du Site Central MVS.

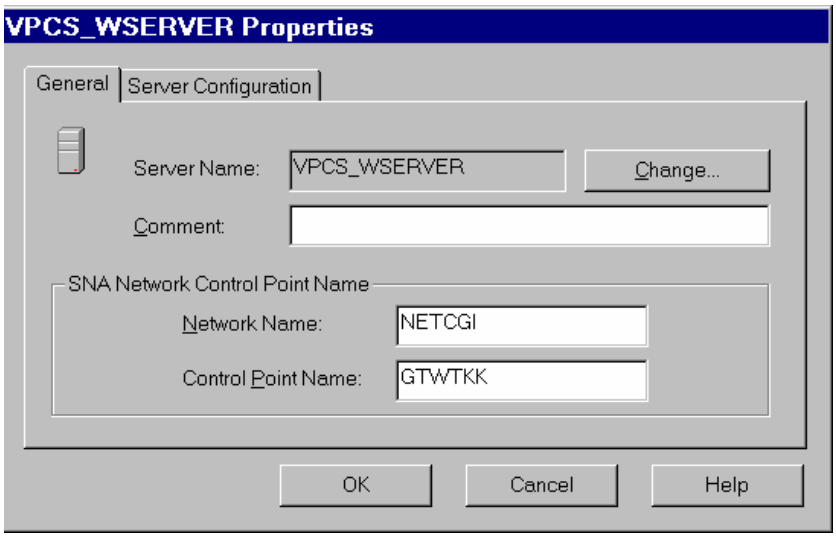

Microsoft SNA Server Manager - [VPCS_WSERVER]
File Edit View Insert Service Tools Window Help
VPCS_WSERVER
VPCS_WSERVER
SNA Servers
VPCS_WSERVER [Active]
Connections
TOKEN1 [Active]
Local APPC LUs
CICS9271
CICS9272
IMS9274
T106TK62
Remote APPC LUs
A01CICS1
A01DSN1
A01IMS62
CICST
Link Services
Print Servers
Shared Folder Servers
TN3270 Servers
TN5250 Servers
Pools
POOL0001
POOLTN01
Configured Users
APPC Modes
CPIC Symbolic Names
Active Users
Host Security Domains
Connections
Local
Connections on 'VPCS_WSERV
TOKEN1
Active
Ready
Domain: VPCS_WSERVER



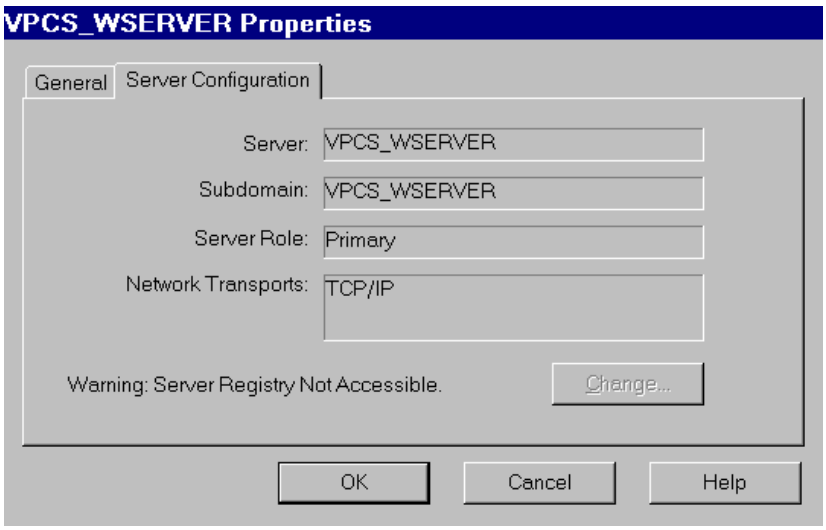
Propriétés du Serveur

`Network Name` = identifiant du réseau SNA (**`NETID`** dans l'**`ATCSTRxx`** de VTAM)

`Control Point Name` = correspondant au **`CPNAME`** dans la définition de la **`PU`** de VTAM



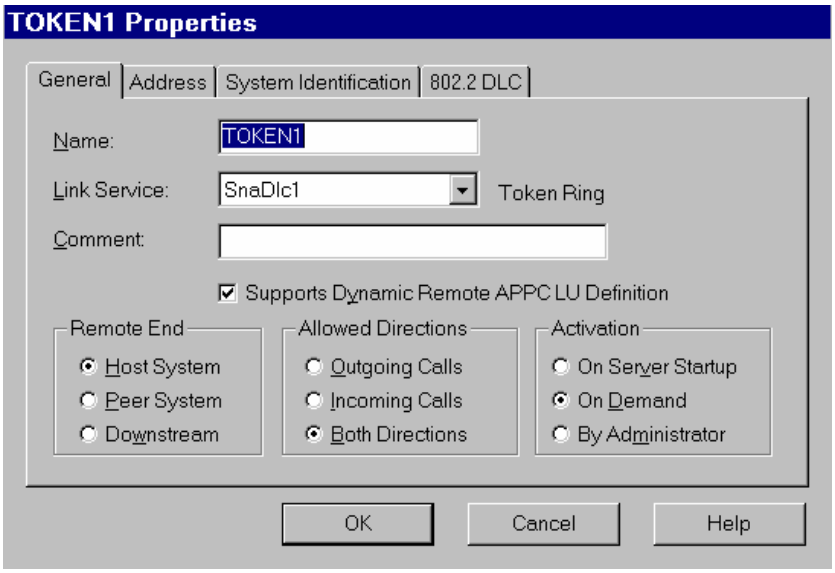
- Type de transport utilisé entre le client et le serveur SNA (ex: protocole TCP/IP)



Propriétés du Lien

`Link Service` : Le type de lien doit être choisi dès l'installation ou installé ultérieurement par le programme Setup. C'est le composant de SNA Server qui communique avec le driver de la carte réseau. Le SNA DLC 802.2 Link service est alloué pour la communication avec le site central dans un réseau LAN Token ring ou Ethernet.

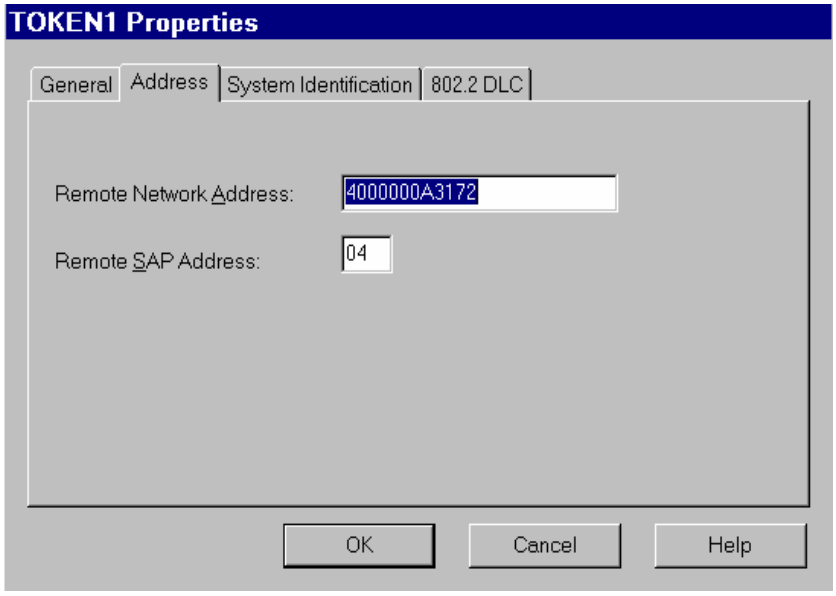
Type de connexion = 802.2



- Définition du TIC contrôleur 3745 dans NCP (attachement au réseau LAN).

```
*********************************************
*            TIC BNN                                  05742090
*********************************************
A01L1TK  LINE ADDRESS=(1088,FULL),
                    PORTADD=01,
                    LOCADD=4000000A3172,
                    ISTATUS=ACTIVE,
                           UACB=(X$P1AX,X$P1AR)
*
A01TK1PU PU ISTATUS=ACTIVE,
                    ADDR=01
*********************************************
```

Remote Network Address = LOCADD (MAC Address) dans la macro **LINE** du **NCP**.



- Paramètres **SSCPNAME** et **NETID** de l'**ATCSTRxx** de VTAM (nécessaire pour la configuration du Remote Node Name dans SNA Server) :

```
*****************************
NOPROMPT,CONFIG=00,SSCPID=01,
MAXSUBA=31,SUPP=NOSUP,
SSCPNAME=A01M,
SSCPORD=DEFINED,
NETID=NETCGI,
HOSTSA=1,
CRPLBUF=(550,,20,,40,40),
IOBUF=(420,182,25,,40,40),
LFBUF=(300,,0,,20,10),
LPBUF=(50,,0,,5,5),
SFBUF=(50,,0,,5,5),
SPBUF=(90,,0,,5,5),
NOTRACE,TYPE=VTAM
*****************************
```

- Définition de la **PU** dans VTAM correspondant à la passerelle SNA Server (utilisée dans la configuration du Local Node Name pour SNA Server) :

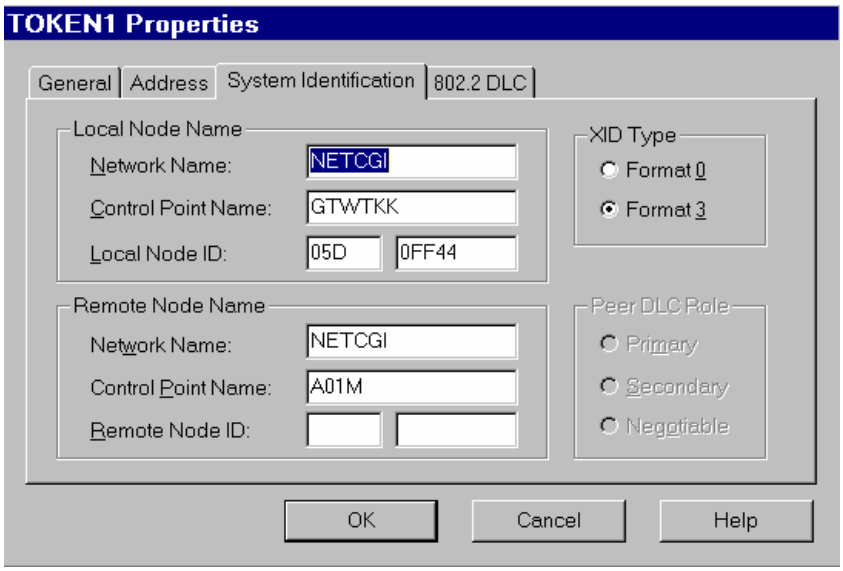
```
*******************************************
*/* SWITCHED MAJOR NODE                                  *
*******************************************
*
SW6TKR   VBUILD TYPE=SWNET,MAXNO=12,MAXGRP=06
*
W6TK00   PU    ADDR=55,
                     CPNAME=GTWTKK,
                     IDBLK=05D,
                     IDNUM=0FF44,
                     DYNLU=YES,
                     MAXPATH=1,
                     DISCNT=YES,
                     IRETRY=YES,
                     VPACING=7,
                     PACING=7,
                     SSCPFM=USSSCS,
                     USSTAB=USSTAB2,
                     MAXDATA=4096,
                     PUTYPE=2,
                     MAXOUT=7,
                     DATMODE=FULL
*
*   ==>       INDEPENDENT  LU
*
CICS9271 LU    LOCADDR=0,
                     ISTATUS=ACTIVE,
                     MODETAB=MTLU62,
                     DLOGMOD=LU62
*
************************************************
```

Local Node Name :

- **Local Node ID = IDBLK & IDNUM**
- **Control Point Name = CPNAME**
- **Network Name = NETID (ATCSTRxx)**

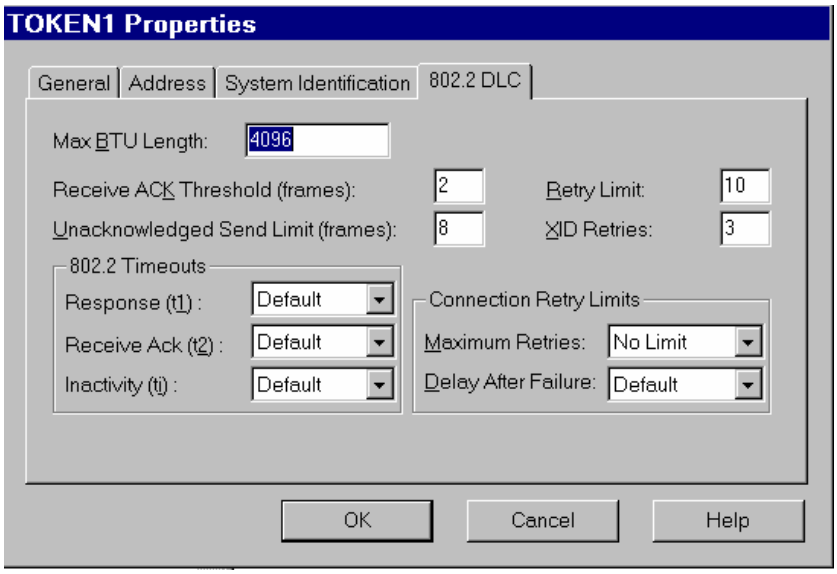
Remote Node Name :

- **Control Point Name = SSCPNAME**
- **Network Name = NETID (ATCSTRxx)**



Max BTU Length (frame size) correspond au **MAXDATA** de la **PU** dans VTAM.

- pour adaptateur Token ring à 4 Mbps, doit être inférieur ou égal à 4195
- pour adaptateur Ethernet, doit être inférieur ou égal à 1493.

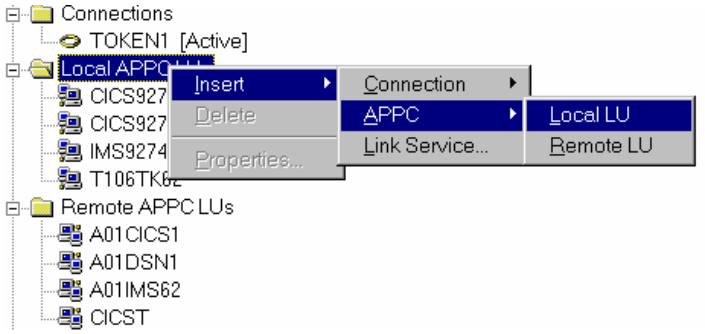


Local APPC LU

APPC utilise une locale LU (indépendante ou dépendante) et une ou plusieurs LUs éloignées. Les sessions APPC communiquent entre deux LUs (Local et Remote).

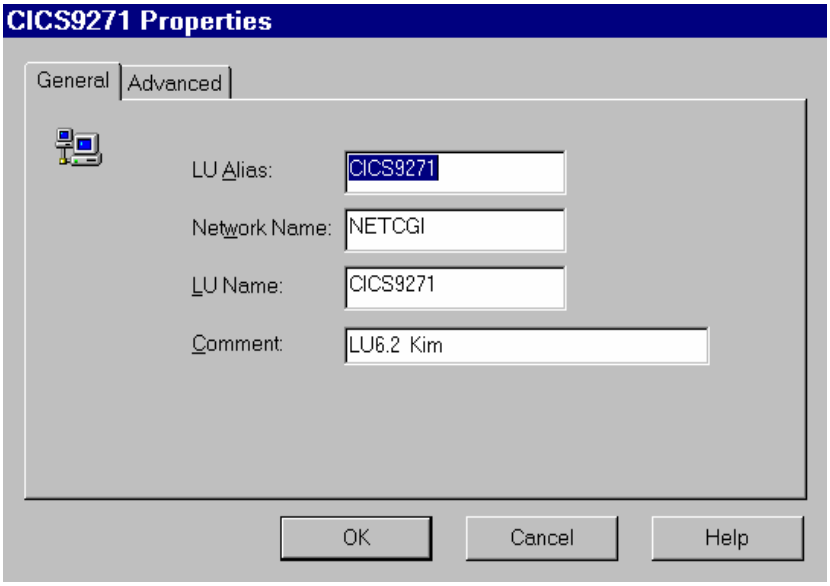
Le minima requis sur MVS pour communiquer avec un TP (Transaction Program) :

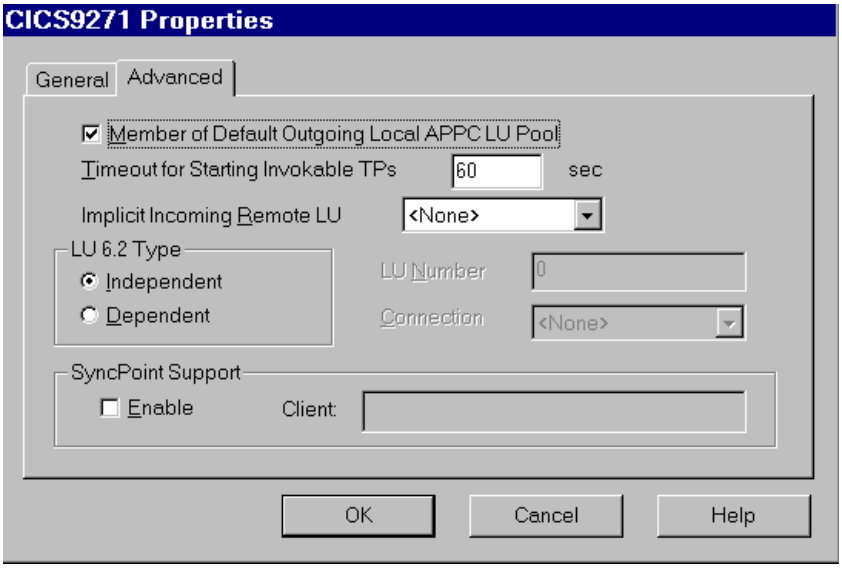
- VTAM version 3.2
- NCP version 5.2 (3745)
- NCP version 4.3 (3725)



`Local LU` est défini avec la **`PU`** dans VTAM comme une LU Indépendante avec **`LOCADDR = 0`**:

- **`LU Name = CICS9271`**
- **`Network Name = NETID dans ATCSTRxx`**
- **`LU Alias =`** Identifiant de la LU en local pour TPs (Transaction Programs)





Remote APPC LU

- Définition de l'APPL CICS dans VTAM :

```
*********************************************************
*/*      THIS MEMBER CONTAINS VTAM APPLICATION DEFINITION
*/*      .      NAME         ACBNAME
*/*             ------------ -------------
*/*      .      A01CICS1     CICST
*********************************************************
A01CICS1 APPL  EAS=160,
                     ACBNAME=CICST,                APPLID FOR ACB
                     AUTH=(ACQ,VPACE,PASS),
                     PARSESS=YES,
                     SONSCIP=YES,
                     MODETAB=MTLU62
*********************************************************
```

- Dans la table SIT du CICS, InterSystem Communication doit être activé : **ISC=YES**

- Définition de la CONNEXION dans CICS :

  Netname (CICS9271) correspond au Local APPC LU

```
------------------------------------------------------------------------
OVERTYPE TO MODIFY                                CICS RELEASE = 0330
CEDA  ALter
  Connection     : SG71
  Group          : GRPISC9
  DEscription  ==> CONNEXION LU6.2 SNA SERVER
 CONNECTION IDENTIFIERS
   Netname      ==> CICS9271
   INDsys       ==>
   REMOTE ATTRIBUTES
   REMOTESystem ==>
   REMOTEName   ==>
 CONNECTION PROPERTIES
 ACcessmethod ==> Vtam
 Protocol     ==> Appc
 SInglesess   ==> No
 DAtastream   ==> User
 RECordformat ==> U
 OPERATIONAL PROPERTIES
 AUtoconnect  ==> Yes
 INService    ==> Yes
 SECURITY
 SEcurityname ==> SYTD
 ATtachsec    ==> Verify
 BINDPassword ==>
 BINDSecurity ==> No
                                                    APPLID=CICST
 ------------------------------------------------------------------------
```

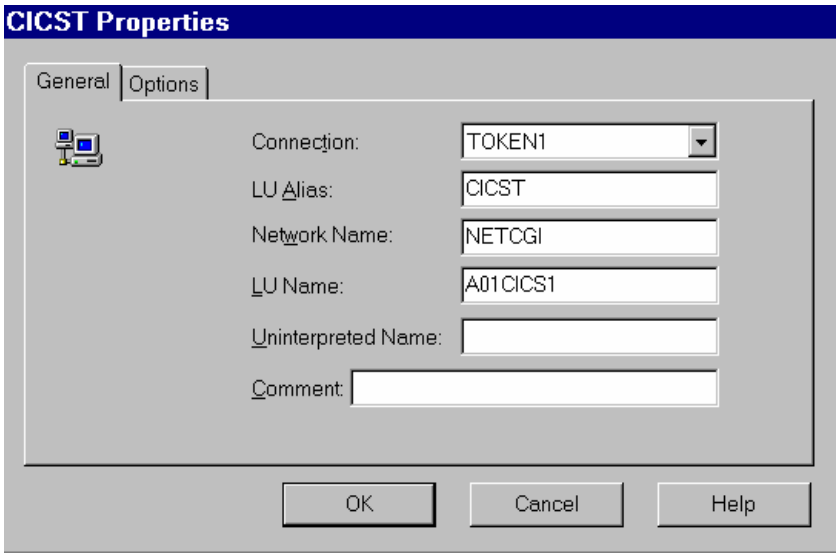
- Définition de la SESSION dans CICS :
  - **Connection (SG71)** correspond au code connexion défini ci-dessus
  - **MOdename (LU62)** correspond au nom du Mode défini dans SNA Server
  - **MAximum (004 , 002)** correspond aux paramètres **Parallel Session Limit** et au **Partner Min Contention Winner** défini dans le Mode (LU62) de SNA Server

- **SENDSize** et **RECEIVESize** correspondent aux paramètres **Max Receive RU Size** et **Max Send RU Size** du Mode (LU62) de SNA Server

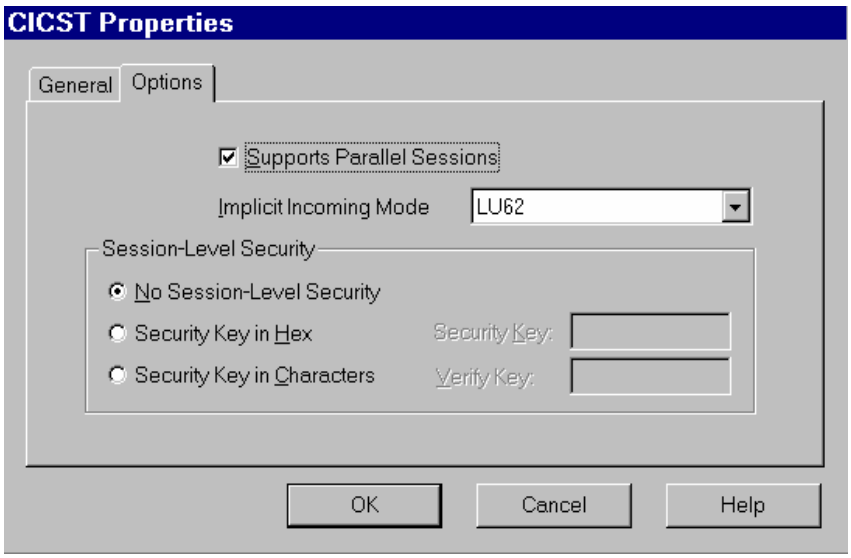
```
------------------------------------------------------------------------
OVERTYPE TO MODIFY                                CICS RELEASE = 0330
CEDA  ALter
Sessions       : SESSIO71
Group          : GRPISC9
DEscription  ==> SESSION LU6.2 SNA Server
SESSION IDENTIFIERS
Connection   ==> SG71
SESSName     ==>
NETnameq     ==>
MOdename     ==> LU62
SESSION PROPERTIES
Protocol     ==> Appc
MAximum      ==> 004 , 002
RECEIVEPfx   ==>
RECEIVECount ==>
SENDPfx      ==>
SENDCount    ==>
SENDSize     ==> 08192
RECEIVESize  ==> 08192
SESSPriority ==> 000
Transaction    :
OPERATOR DEFAULTS
OPERId         :
OPERPriority   : 000
OPERRsl        : 0
OPERSecurity   : 1
PRESET SECURITY
USERId       ==>
OPERATIONAL PROPERTIES
Autoconnect  ==> Yes
INservice      :
Buildchain   ==> Yes
USERArealen  ==> 000
IOarealen    ==> 00000 , 00000
RELreq       ==> Yes
DIscreq      ==> No
NEPclass     ==> 000
RECOVERY
RECOVOption  ==> Sysdefault
RECOVNotify  ==> None

APPLID=CICST
------------------------------------------------------------------------
```

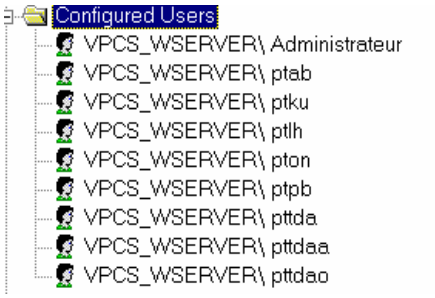
- Définition du Remote APPC LU dans SNA Server:
  - **Network Name = NETID** dans **ATCSTRxx**
  - **LU Alias =** Identifiant de la LU en local TPs (Transaction Programs)
  - **LU Name = APPL ID CICS** dans VTAM (**A01CICS1**)

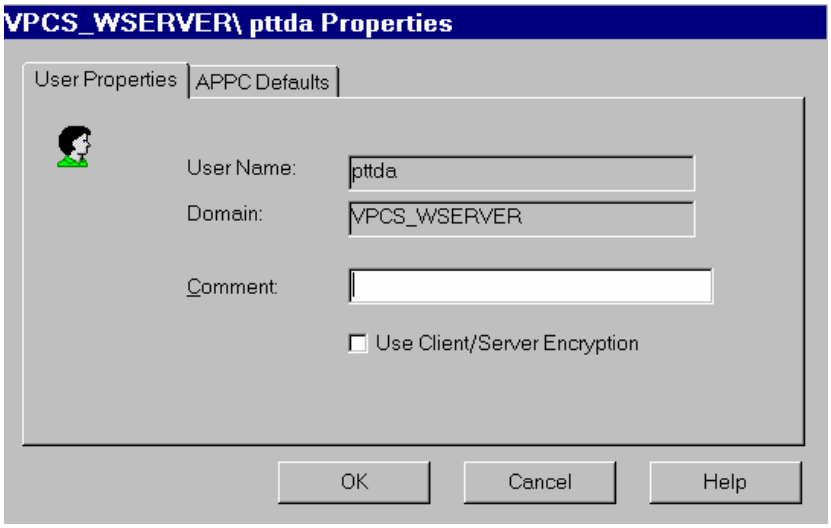


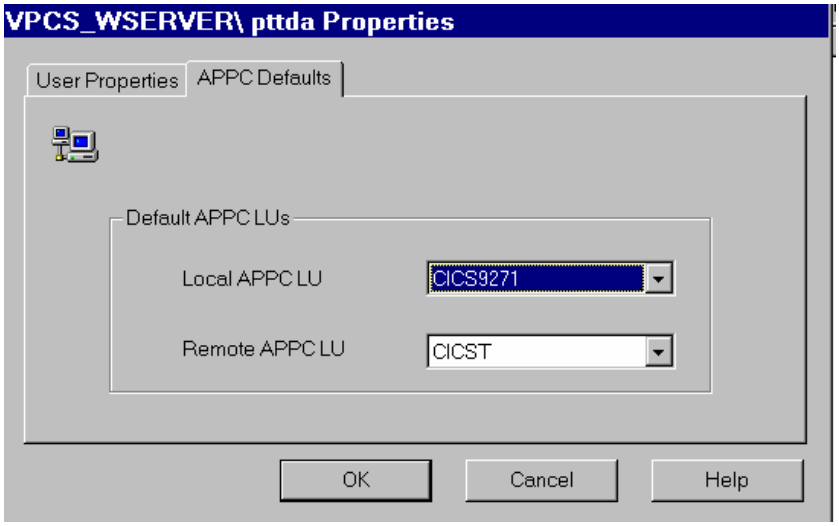
- **`Implicit Incoming Mode = Mode Name`** utilisé pour Supports Parallel Sessions et défini dans l'APPC Modes de SNA Server



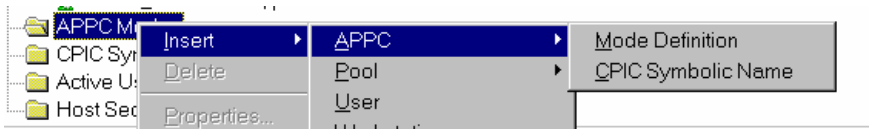
Configuration des Utilisateurs







Définition du MODE



- Définition des MODES LU6.2 dans la table des Modes de VTAM :

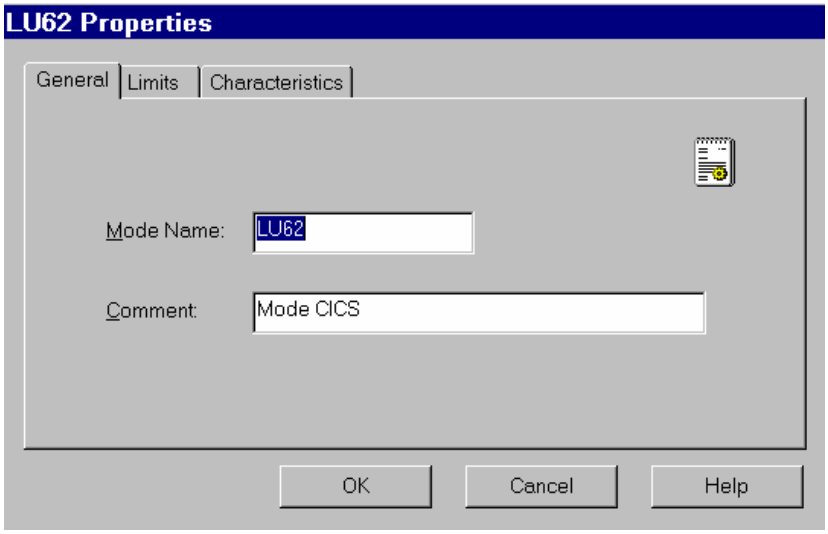
  Le Mode LU62 est un exemple de mode configuré et utilisé avec APPC (LU6.2). Le mode SNASVCMG est inclus dans SNA Server et est utilisé par le "Supports Parallel Sessions".

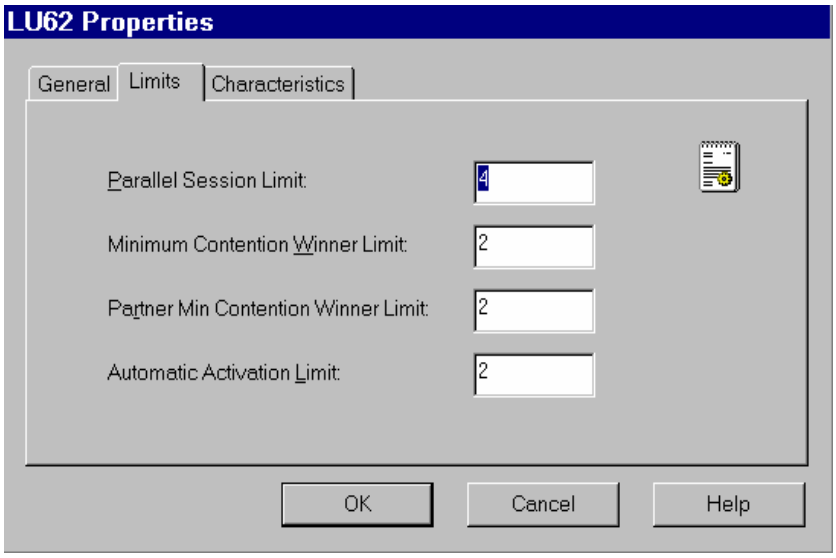
```
------------------------------------------------------------------------
*
***     TITLE '--- "MODTABLE" CONCERNANT LES LU 6.2 ---'    *******
*
SNASVCMG MODEENT LOGMODE=SNASVCMG,
               FMPROF=X'13',
                TSPROF=X'07',
               PRIPROT=X'B0',
                SECPROT=X'B0',
                COMPROT=X'D0B1',
                RUSIZES=X'8585',
```

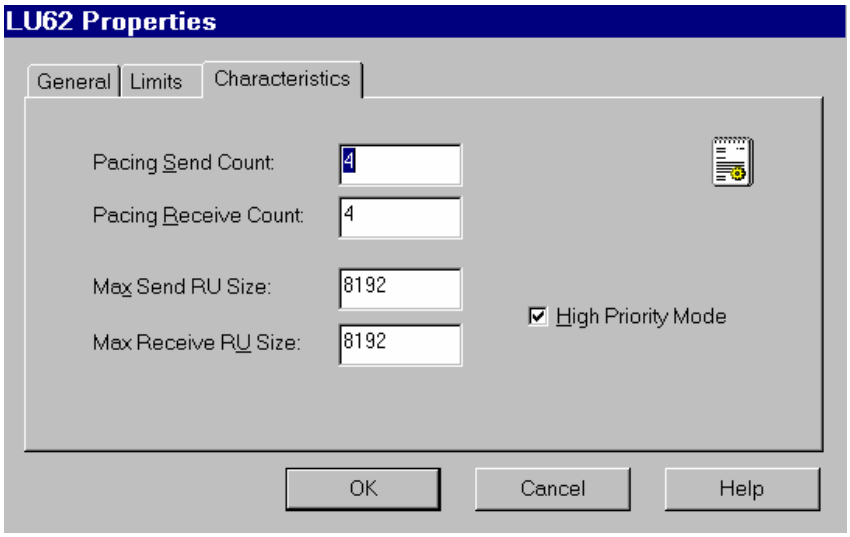
```
                ENCR=B'0000',
                PSERVIC=X'060200000000000000000300'
*
LU62      MODEENT LOGMODE=LU62,
                TYPE=X'00',
                FMPROF=X'13',
                TSPROF=X'07',
                PRIPROT=X'B0',
                SECPROT=X'B0',
                COMPROT=X'50B1',
                RUSIZES=X'8989',
                SRCVPAC=X'00',
                PSNDPAC=X'00',
                SSNDPAC=X'00',
                PSERVIC=X'06020000000000000002C00'
------------------------------------------------------------------------
```

- Définition du Mode dans SNA Server :

  Mode Name = Mode défini dans la table des modes de VTAM



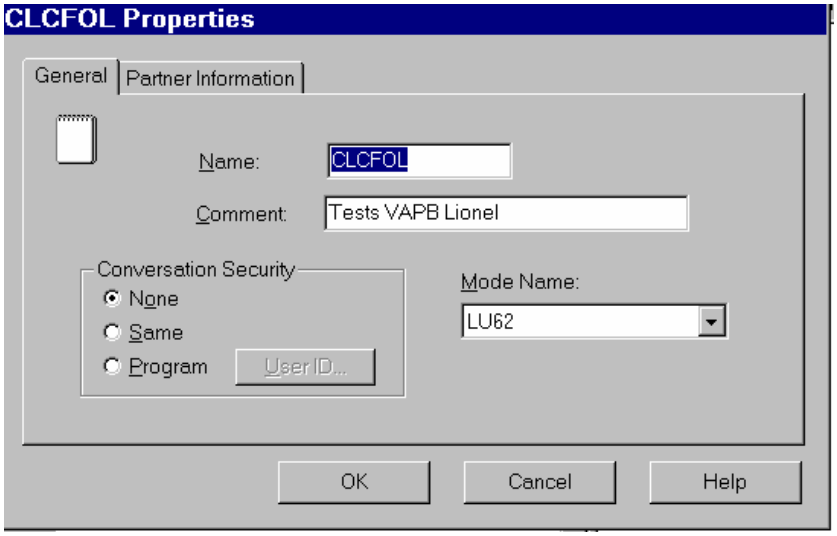




Définition du CPI-C Symbolic Destinations Names (Side Information)

CPI-C est l'interface API de communication. Il fournit un groupe standard d'appels de fonction pour toutes les plates-formes APPC prenant en charge CPI-C.

Le **Name** correspond au nom externe du Moniteur de Communication VA Pac (établit la connexion entre les clients et les programmes serveur utilisant APPC).



- Définition de la transaction CICS :

```
------------------------------------------------------------------------
OVERTYPE TO MODIFY                                  CICS RELEASE = 0330
CEDA  ALter
TRansaction     : VP20
Group           : VISUAL
DEscription  ==> VISUAL/CLIENT WIN/NT CPIC
PROGram      ==> CLCFOL
TWasize      ==> 00000
PROFile      ==> DFHCICST
PArtitionset ==>
STatus       ==> Enabled
PRIMedsize      : 00000
TASKDATALoc  ==> Below
```

```
TASKDATAKey  ==> User
REMOTE ATTRIBUTES
DYnamic      ==> No
REMOTESystem ==>
REMOTEName   ==>
TRProf       ==>
Localq       ==>
SCHEDULING
PRIOrity     ==> 001
TClass       ==> No
ALIASES
Alias        ==>
TASKReq      ==>
XTRanid      ==>
TPName       ==>
       ==>
 XTPname      ==>
                      ==>
                      ==>
RECOVERY
DTimout      ==> No
Indoubt      ==> Backout
RESTart      ==> No
SPurge       ==> No
TPUrge       ==> No
DUmp         ==> Yes
TRACe        ==> Yes
SECURITY
RESSec       ==> No
Cmdsec       ==> No
Extsec         : No
TRANsec        : 01
RSl            : 00

                                                                APPLID=CICST
-------------------------------------------------------------------------------
```

Si un programme utilise DB2 au cours de la transaction, il est nécessaire de rattacher le plan DB2 à la transaction du moniteur serveur en la déclarant dans la table RCT du CICS/ESA :

```
DSNCRCT TYPE=ENTRY,TXID=(VP20),THRDM=2,
THRDA=2,PLAN=VP20,AUTH=(USERID,*,*)
```

Si l'application fait des accès DB2, il est nécessaire d'autoriser et de rattacher la transaction utilisateur au plan DB2 dans la table RCT de la région CICS :

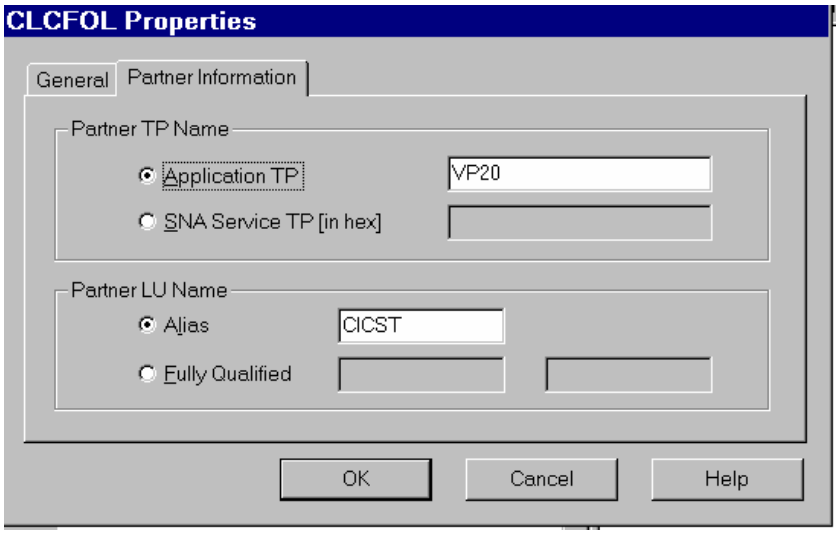
```
DSNCRCT TYPE=ENTRY,TXID=(VP20),THRDM=6,
THRDA=6,PLAN=VP20,AUTH=(USERID,*,*)
```

- Définition du **Partner TP Name** :

  Application TP = Code Transaction activant le Moniteur de Communication sur le Host et défini ci-dessus

- Définition du **Partner LU Name** :

  Alias = Alias défini dans Remote APPC LU


CLCFOL Properties
General
Partner Information
Partner TP Name
Application TP
VP20
SNA Service TP [in hex]
Partner LU Name
Alias
CICST
Fully Qualified
OK
Cancel
Help

CICS ECI

Configuration MVS et CICS

Définitions VTAM

Minima requis : VTAM Version 3.3

Définition de l'ATCSTR de VTAM

```
*******************************
NOPROMPT,CONFIG=00,SSCPID=01,
MAXSUBA=31,SUPP=NOSUP,
SSCPNAME=A01M,
SSCPORD=DEFINED,
NETID=NETCGI,
HOSTSA=1,
CRPLBUF=(550,,20,,40,40),
IOBUF=(420,182,25,,40,40),
LFBUF=(300,,0,,20,10),
LPBUF=(50,,0,,5,5),
SFBUF=(50,,0,,5,5),
SPBUF=(90,,0,,5,5),
NOTRACE,TYPE=VTAM
*******************************
```

MAC Address du TIC (attachement LAN) contrôleur 3745 dans NCP :

```
*********************************************
*            TIC BNN                                  05742090
*********************************************
A01L1TK  LINE ADDRESS=(1088,FULL),
              PORTADD=01,
              LOCADD=400003172000,
              ISTATUS=ACTIVE,
              UACB=(X$P1AX,X$P1AR)
*
A01TK1PU PU ISTATUS=ACTIVE,
               ADDR=01
*********************************************
```

Définition du CICS

```
A01CICS1 APPL  EAS=160,                ESTIMATED CONCURRENT SESSIONS   *
               ACBNAME=CICST,          APPLID FOR ACB                  *
               AUTH=(ACQ,VPACE,PASS),  CICS CAN ACQUIRE & PASS TMLS    *
               PARSESS=YES,                   Supports parallel Sessions
               SONSCIP=YES,                                           *
               MODETAB=MTLU62                 nom de la table des modes
```

Définition du mode

- Définition des caractéristiques pour les Sessions LU6.2
- Le Mode SNASVCMG est utilisé avec le support "Parallel Sessions".

```
 TITLE '--- "MODTABLE" CONCERNANT LES LU 6.2 ---'
 *
 MTLU62   MODETAB
          SPACE 4
 SNASVCMG MODEENT LOGMODE=SNASVCMG, FMPROF=X'13',
                TSPROF=X'07', PRIPROT=X'B0', SECPROT=X'B0',
                COMPROT=X'D0B1', RUSIZES=X'8585', ENCR=B'0000',
                PSERVIC=X'060200000000000000000300'
 LU62     MODEENT LOGMODE=LU62, TYPE=X'00', FMPROF=X'13',
                TSPROF=X'07', PRIPROT=X'B0', SECPROT=X'B0',
                COMPROT=X'50B1', RUSIZES=X'8787', SRCVPAC=X'00',
```

```
                PSNDPAC=X'00', SSNDPAC=X'00',
                PSERVIC=X'060200000000000000002C00'
```

Définition de la passerelle SNA

```
*/* LIB: SYS1.VTAMLST(SW1TKR)
*/*
*/*
*/* SWITCHED MAJOR NODE TOKEN-RING ST-MARC :            -  06/07/95.
*/*   --->    LIEN XCA MAJOR NODE ==> XCA1TKR  (IBM3172-3)
*/*   --->    LIEN GROUPE XCA     ==> GRP02
*             -------------------------------
*    - MODEL FOR IDBLK X'05D' - OS/2 COMMUNICATIONS MANAGER
*             -------------------------------
*
SW1TKR   VBUILD TYPE=SWNET,MAXNO=99,MAXGRP=10
*
*          ----------------------------------------
* -------> DEFINING A GATEWAY TOKEN-RING --> GTWTK1  <--------
*          ----------------------------------------
W1TK00   PU    ADDR=50,
               CPNAME=GTWTK1,
               IDBLK=05D,
               IDNUM=00002,
               DYNLU=YES,
               MAXPATH=1,
               DISCNT=NO,
               IRETRY=YES,
               VPACING=7,
               PACING=7,
               SSCPFM=USSSCS,
               MAXDATA=4096,
               PUTYPE=2
```

L'option **DYNLU=YES** permet d'éviter toute définition complémentaire de Lu 6.2 au niveau VTAM (pour les postes du réseau TR).

Définition d'une LU indépendante

Malgré ce qui vient d'être dit ci-dessus, il peut être intéressant de définir une LU indépendante pour les premiers tests de communication :

```
*/* LIB: SYS1.VTAMLST(SW1TKR)
*/*
*/*
*/* SWITCHED MAJOR NODE TOKEN-RING ST-MARC :            -  06/07/95.
*/*
*/*
*  INDEPENDENT LUS
*
CGI5075          LU      LOCADDR=0,
                 ISTATUS=ACTIVE,
                 DLOGMOD=LU62,
                 MODETAB=MTLU62
```

Définitions APPC/MVS

Minima requis : MVS/ESA Version 4.2

Il n'y a pas de définitions spécifiques car on utilise la couche APPC livrée avec la version de CICS.

Définitions CICS

Minima requis : CICS/ESA Version 4.1

Paramètre de l'InterSystem Communication dans la table SIT

ISC=YES

Connexion

```
 Connection     : SGFB
 Group          : GRPISC5
 DEscription    :
CONNECTION IDENTIFIERS
 Netname        : CGI5075       à déclarer comme Local LU dans CM/2
                                (codification libre)
 INDsys         :
REMOTE ATTRIBUTES
 REMOTESystem   :
 REMOTEName     :
CONNECTION PROPERTIES
 ACcessmethod   : Vtam
 Protocol       : Appc
 SInglesess     : No             dans le cas d'une LU indépendante
 DAtastream     : User
 RECordformat   : U
OPERATIONAL PROPERTIES
 AUtoconnect    : Yes
 INService      : Yes
SECURITY
 SEcurityname   : PTPD
 ATtachsec      : Verify         Vérification du userid et password à la
                                conversation
 BINDPassword   :
 BINDSecurity   : No
```

Session

```
 Sessions       : SESSIOFB
 Group          : GRPISC5
 DEscription    :
SESSION IDENTIFIERS
 Connection     : SGFB           code de la connexion définie ci-dessus
 SESSName       :
 NETnameq       :
 MOdename       : LU62           mode défini dans la MODTABLE de VTAM
SESSION PROPERTIES
 Protocol       : Appc
 MAximum        : 004 , 002
 RECEIVEPfx     :
 RECEIVECount   :
 SENDPfx        :
 SENDCount      :
 SENDSize       : 08192
 RECEIVESize    : 08192
 SESSPriority   : 000
 Transaction    :
OPERATOR DEFAULTS
 OPERId         :
 OPERPriority   : 000
 OPERRsl        : 0
 OPERSecurity   : 1
PRESET SECURITY
 USERId         :
OPERATIONAL PROPERTIES
 Autoconnect    : Yes
 INservice      :
 Buildchain     : Yes
 USERArealen    : 000
 IOarealen      : 00000 , 00000
 RELreq         : Yes
 DIscreq        : No
 NEPclass       : 000
RECOVERY
 RECOVOption    : Sysdefault
 RECOVNotify    : None
```

Définition de la transaction

Définition d'une transaction Miroir utilisateur

```
------------------------------------------------------
                                      CICS RELEASE = 0410
   TRANSaction     : VEC1
   Group           : VPCS250
   DEscription     :
   PROGram         : DFHMIRS
   TWasize         : 00000              0-32767
   PROFile         : DFHCICSA
   PArtitionset    :
   STAtus          : Enabled            Enabled ! Disabled
   PRIMedsize      : 00000              0-65520
   TASKDATALoc     : Below              Below   ! Any
   TASKDATAKey     : User               User    ! Cics
   STOrageclear    : No                 No      ! Yes
   RUnaway         : System             System  ! 0-2700000
   SHutdown        : Disabled           Disabled ! Enabled
   ISolate         : Yes                Yes     ! No
  REMOTE ATTRIBUTES
+  DYnamic         : No                 No ! Yes
REMOTESystem    :
 REMOTEName     :
 TRProf         :
 Localq         :                    No ! Yes
SCHEDULING
 PRIOrity       : 001                0-255
 TClass         : No                 No ! 1-10
 TRANClass      : DFHTCL00
ALIASES
 Alias          :
 TASKReq        :
 XTRanid        :
 TPName         :
                :
 XTPname        :
                :
                :
RECOVERY
  DTimout        : No                 No ! 1-6800
  INdoubt        : Backout            Backout ! Commit ! Wait
  RESTart        : No                 No ! Yes
  SPurge         : No                 No ! Yes
  TPUrge         : No                 No ! Yes
  DUmp           : Yes                Yes ! No
  TRACe          : Yes                Yes ! No
  COnfdata       : No                 No ! Yes
 SECURITY
  RESSec         : Yes                No ! Yes
  CMdsec         : Yes                No ! Yes
  Extsec         : No
  TRANSec        : 01                 1-64
  RSl            : 00                 0-24 ! Public
------------------------------------------------------------
```

De plus, si une transaction fait des accès à une base DB2, il faut autoriser et rattacher celle-ci au plan DB2. Le code transaction et le plan DB2 doivent donc être déclaré dans la Table RCT. Par défaut le transaction CPMI (Transaction Miroir) est utilisée avec le CICS Client. Dans cet exemple le plan DB2 utilisé avec application Cliente se nomme ATDF :

```
DSNRCT TYPE = ENTRY,TXID=(CPMI,CSPM,VICL),
         THRDM=2,THRDA=2,PLAN=ATDF,AUTH=(USERID,*,*)
```

Si la transaction miroir CPMI provoque un Abend ACN1, c'est que la table de conversion DFHCNV n'est pas définie dans la région CICS. Elle est nécessaire pour l'utilisation de cette transaction. Définir cette table puis assembler et LinkEdit avec les paramètres suivants :

```
*********************************************************************
//EXBCCNV  JOB (009),'BC',CLASS=X,MSGCLASS=X,NOTIFY=SYTD
//CICSCNV  EXEC DFHAUPLE,
// PARM.LNKEDT='RENT,REUS,LIST,XREF,LET,NCAL,AMODE=31,RMODE=ANY'
//ASM.SYSLIB DD DSN=CICS330.SDFHMAC,DISP=SHR
//ASSEM.SYSUT1 DD DSN=PT$EXP.CICST330.SOURCE(DFHCNV),DISP=SHR
//LNKEDT.SYSLMOD DD DSN=PT$PDV.PB80204.MTR8,DISP=SHR

DFHCNV   TYPE=INITIAL
DFHCNV   TYPE=ENTRY,RTYPE=PC,RNAME=TESTVP,CLINTCP=(850,437),  *
         SRVERCP=297
DFHCNV   TYPE=SELECT,OPTION=DEFAULT
DFHCNV   TYPE=FIELD,OFFSET=0,DATATYP=CHARACTER,DATALEN=8051,  *ES
         LAST=YES
DFHCNV   TYPE=FINAL
END
*********************************************************************
```

CICS TCP/IP Sockets Interface

Configuration CICS TCP/IP

Prérequis

MVS/ESA :

TCP/IP Version 3, Release 1

CICS/ESA Version 3, Release 3

CICS TCP/IP Socket Interface Version 3.1

CICS Startup

Modification du JCL de lancement pour la région CICS.

```
------------------------------------------------------------------------
//PMTCICST  JOB (008),'CICS TEST PAC',MSGLEVEL=(2,0),CLASS=O,
//   MSGCLASS=X
//CICST    PROC INDEX=CICS330,
//             UTINDX='PT$EXP.CICST330',
//             REGSZE=6M,
//             START='COLD',
//             SIP=T,
//DFHRPL   DD  DSN=&UTINDX..LNK,DISP=SHR
//                       DD  DSN=SYS1.TCPIP310.SEZALINK,DISP=SHR
//                       DD  DSN=TCPIP310.SEZATCP,DISP=SHR
//TCPDATA   DD SYSOUT=&OUTC,DCB=(DSORG=PS,RECFM=V,BLKSIZE=136)
------------------------------------------------------------------------
```

Définition transactions CICS TCP/IP

- Listener Task

```
-----------------------------------------------------------------
TRansaction    : CSKL
  Group          : TCPIPI
  DEscription    : Listener Task
  PROGram        : EZACIC02
  TWasize        : 00000
  PROFile        : DFHCICST
  PArtitionset   :
  STatus         : Enabled
  PRIMedsize     : 00000
  TASKDATALoc    : Below
  TASKDATAKey    : Cics
 REMOTE ATTRIBUTES
  DYnamic        : No
  REMOTESystem   :
  REMOTEName     :
  TRProf         :
  Localq         :
-----------------------------------------------------------------
```

*TCP/IP Version 3.1.0

- Enable the Socket Interface

```
-----------------------------------------------------------------
TRansaction    : CSKE
  Group          : TCPIPI
  DEscription    : Enable Sockets Interface
  PROGram        : EZACIC00
  TWasize        : 00000
  PROFile        : DFHCICST
  PArtitionset   :
  STatus         : Enabled
  PRIMedsize     : 00000
  TASKDATALoc    : Below
```

```
  TASKDATAKey    : Cics
 REMOTE ATTRIBUTES
  DYnamic        : No
  REMOTESystem   :
  REMOTEName     :
  TRProf         :
  Localq         :
--------------------------------------------------------------------
```

- Terminate the socket interface

```
--------------------------------------------------------------------
TRansaction    : CSKD
   Group          : TCPIPI
   DEscription    : Disable Sockets Interface
   PROGram        : EZACIC00
   TWasize        : 00000
   PROFile        : DFHCICST
   PArtitionset   :
   STatus         : Enabled
   PRIMedsize     : 00000
   TASKDATALoc    : Below
   TASKDATAKey    : Cics
  REMOTE ATTRIBUTES
   DYnamic        : No
   REMOTESystem   :
   REMOTEName     :
   TRProf         :
+  Localq         :
--------------------------------------------------------------------
```

*TCP/IP Version 3.2.0

- Configure the socket interface

```
--------------------------------------------------------------------
TRansaction    : EZAC
  Group          : TCPIPI
  DEscription    : CONFIGURE SOCKETS INTERFACE
  PROGram        : EZACIC23

  TWasize        : 00000
  PROFile        : DFHCICST
  PArtitionset   :
  STatus         : Enabled
  PRIMedsize     : 00000
  TASKDATALoc    : Below
  TASKDATAKey    : Cics
 REMOTE ATTRIBUTES
  DYnamic        : No
  REMOTESystem   :
  REMOTEName     :
  TRProf         :
  Localq         :
--------------------------------------------------------------------
```

- Enable the socket interface

```
--------------------------------------------------------------------
TRansaction    : EZAO
  Group          : TCPIPI
  DEscription    : ENABLE SOCKETS INTERFACE
  PROGram        : EZACIC00

  TWasize        : 00000
  PROFile        : DFHCICST
  PArtitionset   :
  STatus         : Enabled
  PRIMedsize     : 00000
  TASKDATALoc    : Below
  TASKDATAKey    : Cics
 REMOTE ATTRIBUTES
  DYnamic        : No
  REMOTESystem   :
  REMOTEName     :
  TRProf         :
```

```
  Localq         :
-----------------------------------------------------------------
```

- Terminate the socket interface

```
-----------------------------------------------------------------
TRansaction    : EZAP
   Group          : TCPIPI
   DEscription    : DISABLE SOCKETS INTERFACE
   PROGram        : EZACIC22
   TWasize        : 00000
   PROFile        : DFHCICST
   PArtitionset   :
   STatus         : Enabled
   PRIMedsize     : 00000
   TASKDATALoc    : Below
   TASKDATAKey    : Cics
  REMOTE ATTRIBUTES
   DYnamic        : No
   REMOTESystem   :
   REMOTEName     :
   TRProf         :
-----------------------------------------------------------------
```

Définition programmes CICS TCP/IP

```
-------------------------------------------------------------------------------
PROGram        : EZACIC00
    Group          : TCPIPI
    DEscription    : Connection Manager
    Language       : Assembler
    RELoad         : No
    RESident       : No
    USAge          : Transient
    USElpacopy     : No
    Status         : Enabled
    RSl            : 00
    Cedf           : Yes
    DAtalocation   : Any
    EXECKey        : CICS
   REMOTE ATTRIBUTES
    REMOTESystem   :
 +  REMOTEName     :
      Transid        :
  EXECUtionset   : Fullapi
-------------------------------------------------------------------------------

-------------------------------------------------------------------------------
PROGram        : EZACIC02
   Group          : TCPIPI
   DEscription    : Listener
   Language       : Assembler
   RELoad         : No
   RESident       : Yes
   USAge          : Normal
   USElpacopy     : No
   Status         : Enabled
   RSl            : 00
   Cedf           : Yes
   DAtalocation   : Any
   EXECKey        : CICS
  REMOTE ATTRIBUTES
   REMOTESystem   :
+  REMOTEName     :
Transid        :
  EXECUtionset   : Fullapi
-------------------------------------------------------------------------------

-------------------------------------------------------------------------------
Mapset         : EZACICM
   Group          : TCPIPI
   Description    : Mapset for Connection Manager
```

```
   REsident       : No
   USAge          : Transient
   USElpacopy     : No
   Status         : Enabled
   RSl            : 00
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
PROGram        : EZACIC01
    Group          : TCPIPI
    DEscription    : Task Related User Exit
    Language       : Assembler
    RELoad         : No
    RESident       : Yes
    USAge          : Normal
    USElpacopy     : No
    Status         : Enabled
    RSl            : 00
    Cedf           : Yes
    DAtalocation   : Any
    EXECKey        : CICS
   REMOTE ATTRIBUTES
    REMOTESystem   :
 +  REMOTEName     :
Transid        :
 EXECUtionset   : Fullapi
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
PROGram        : EZACIC20
    Group          : TCPIPI
    DEscription    : Initialization/termination for CICS Sockets
    Language       : Assembler
    RELoad         : No
    RESident       : No
    USAge          : Transient
    USElpacopy     : No
    Status         : Enabled
    RSl            : 00
    Cedf           : Yes
    DAtalocation   : Any
    EXECKey        : CICS
   REMOTE ATTRIBUTES
    REMOTESystem   :
 +  REMOTEName     :
Transid        :
 EXECUtionset   : Fullapi
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
PROGram        : EZACIC21
    Group          : TCPIPI
    DEscription    : Initialization Module for CICS Sockets
    Language       : Assembler
    RELoad         : No
    RESident       : No
    USAge          : Transient
    USElpacopy     : No
    Status         : Enabled
    RSl            : 00
    Cedf           : Yes
    DAtalocation   : Any
    EXECKey        : CICS
   REMOTE ATTRIBUTES
    REMOTESystem   :
 +  REMOTEName     :
Transid        :
 EXECUtionset   : Fullapi
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
PROGram        : EZACIC22
    Group          : TCPIPI
    DEscription    : Termination Module for CICS Sockets
    Language       : Assembler
    RELoad         : No
```

```
    RESident       : No
    USAge          : Transient
    USElpacopy     : No
    Status         : Enabled
    RSl            : 00
    Cedf           : Yes
    DAtalocation   : Any
    EXECKey        : CICS
   REMOTE ATTRIBUTES
    REMOTESystem   :
 +  REMOTEName     :
Transid        :
 EXECUtionset   : Fullapi
------------------------------------------------------------------------------

------------------------------------------------------------------------------
PROGram        : EZACIC23
    Group          : TCPIPI
    DEscription    : Primary Module for Transaction EZAC
    Language       : Assembler
    RELoad         : No
    RESident       : No
    USAge          : Transient
    USElpacopy     : No
    Status         : Enabled
    RSl            : 00
    Cedf           : Yes
    DAtalocation   : Any
    EXECKey        : User
   REMOTE ATTRIBUTES
    REMOTESystem   :
 +  REMOTEName     :
Transid        :
 EXECUtionset   : Fullapi
------------------------------------------------------------------------------

------------------------------------------------------------------------------
PROGram        : EZACIC24
    Group          : TCPIPI
    DEscription    : Message Delivery Module for CICS Sockets
    Language       : Assembler
    RELoad         : No
    RESident       : No
    USAge          : Transient
    USElpacopy     : No
    Status         : Enabled
    RSl            : 00
    Cedf           : Yes
    DAtalocation   : Any
    EXECKey        : CICS
   REMOTE ATTRIBUTES
    REMOTESystem   :
 +  REMOTEName     :
Transid        :
 EXECUtionset   : Fullapi
------------------------------------------------------------------------------

------------------------------------------------------------------------------
PROGram        : EZACIC25
    Group          : TCPIPI
    DEscription    : Cache Module for the Domain Name Server
    Language       : Assembler
    RELoad         : No
    RESident       : No
    USAge          : Normal
    USElpacopy     : No
    Status         : Enabled
    RSl            : 00
    Cedf           : Yes
    DAtalocation   : Any
    EXECKey        : User
   REMOTE ATTRIBUTES
    REMOTESystem   :
 +  REMOTEName     :
Transid        :
 EXECUtionset   : Fullapi
```

```
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
PROGram        : EZACICME
    Group          : TCPIPI
    DEscription    : US English Text Delivery Module
    Language       : Assembler
    RELoad         : No
    RESident       : No
    USAge          : Normal
    USElpacopy     : No
    Status         : Enabled
    RSl            : 00
    Cedf           : Yes
    DAtalocation   : Any
    EXECKey        : CICS
   REMOTE ATTRIBUTES
    REMOTESystem   :
 +  REMOTEName     :
Transid        :
 EXECUtionset   : Fullapi
--------------------------------------------------------------------------------
```

Définition Table DCT

Définition d'une queue data transitoire TCPM pour le listener, dans la table DCT.

```
--------------------------------------------------------------------------------
TCPDATA  DFHDCT TYPE=SDSCI,        TCP/IP OUTPUT
                BLKSIZE=136,
                BUFNO=1,
                DSCNAME=TCPDATA,
                RECSIZE=132,
                RECFORM=VARUNBA,
                TYPEFLE=OUTPUT
*
TCPM     DFHDCT TYPE=EXTRA,        USED FOR MESSAGES - SEE
                DESTID=TCPM,                  INDDEST=TCPM BELOW
                DSCNAME=TCPDATA
*
TCPIN    DFHDCT TYPE=INTRA,        TCP/IP
                DESTID=TRAA,
                DESTFAC=FILE,
                TRIGLEV=1,
                TRANSID=TRAA
--------------------------------------------------------------------------------
```

Définitions et initialisations des fichiers de Configuration (*TCP/IP Version 3.2.0)

EZACONFG : Fichier de configuration Sockets CICS

```
--------------------------------------------------------------------------------
File           : EZACONFG
 Group          : SOCKETS
 DEScription    : CICS SOCKETS CONFIGURATION FILE
VSAM PARAMETERS
 DSNAme         : CICS.STM9.SOCKETS.CFG
 Password       :                  PASSWORD NOT SPECIFIED
 Lsrpoolid      : 1                1-8 ! None
 DSNSharing     : Allreqs          Allreqs ! Modifyreqs
 STRings        : 001              1-255
 Nsrgroup       :
REMOTE ATTRIBUTES
 REMOTESystem   :
 REMOTEName     :
 RECORDSize     :                  1-32767
 Keylength      :                  1-255
INITIAL STATUS
 STAtus         : Enabled          Enabled ! Disabled ! Unenabled
Opentime       : Startup          Firstref ! Startup
 DIsposition    : Share            Share ! Old
BUFFERS
 DAtabuffers    : 00002            2-32767
```

```
 Indexbuffers    : 00001              1-32767
DATATABLE PARAMETERS
 Table           : No                 No ! Cics ! User
 Maxnumrecs      :                    16-16777215
DATA FORMAT
 RECORDFormat    : V                  V ! F
OPERATIONS
 Add             : No                 No ! Yes
 BRowse          : Yes                No ! Yes
 DELete          : No                 No ! Yes
 REAd            : Yes                Yes ! No
 Update          : No                 No ! Yes
AUTO JOURNALLING
JOurnal         : No                 No ! 1-99
 JNLRead         : None               None ! Updateonly ! Readonly ! All
 JNLSYNCRead     : No                 No ! Yes
 JNLUpdate       : No                 No ! Yes
 JNLAdd          : None               None ! Before ! AFter ! ALl
 JNLSYNCWrite    : No                 Yes ! No
RECOVERY PARAMETERS
 RECOVery        : None               None ! Backoutonly ! All
 Fwdrecovlog     : No                 No ! 1-99
 BAckuptype      : Static             Static ! Dynamic
SECURITY
 RESsecnum       : 00                 0-24 ! Public
-------------------------------------------------------------------------------
```

EZACACHE : Fichier qui est nécessaire si vous utilisez la fonction du Cache Domain Name Server

```
-------------------------------------------------------------------------------
File            : EZACACHE
 Group           : SOCKETS
 DEScription     : DOMAIN NAME SERVER CACHE CONFIGURATION FILE
VSAM PARAMETERS
 DSNAme          : CICS.STM9.SOCKETS.EZACACHE
 Password        :                    PASSWORD NOT SPECIFIED
 Lsrpoolid       : 1                  1-8 ! None
 DSNSharing      : Allreqs            Allreqs ! Modifyreqs
 STRings         : 020                1-255
 Nsrgroup        :
REMOTE ATTRIBUTES
 REMOTESystem    :
 REMOTEName      :
 RECORDSize      :                    1-32767
 Keylength       :                    1-255
INITIAL STATUS
 STAtus          : Enabled            Enabled ! Disabled ! Unenabled
Opentime        : Startup            Firstref ! Startup
 DIsposition     : Old                Share ! Old
BUFFERS
 DAtabuffers     : 00060              2-32767
 Indexbuffers    : 02000              1-32767
DATATABLE PARAMETERS
 Table           : User               No ! Cics ! User
 Maxnumrecs      : 00004000           16-16777215
DATA FORMAT
 RECORDFormat    : V                  V ! F
OPERATIONS
 Add             : Yes                No ! Yes
 BRowse          : Yes                No ! Yes
 DELete          : Yes                No ! Yes
 REAd            : Yes                Yes ! No
 Update          : Yes                No ! Yes
AUTO JOURNALLING
JOurnal         : No                 No ! 1-99
  JNLRead         : None               None ! Updateonly ! Readonly ! All
  JNLSYNCRead     : No                 No ! Yes
  JNLUpdate       : No                 No ! Yes
  JNLAdd          : None               None ! Before ! AFter ! ALl
  JNLSYNCWrite    : No                 Yes ! No
 RECOVERY PARAMETERS
  RECOVery        : None               None ! Backoutonly ! All
  Fwdrecovlog     : No                 No ! 1-99
  BAckuptype      : Static             Static ! Dynamic
```

```
 SECURITY
  RESsecnum      : 00                 0-24 ! Public
-------------------------------------------------------------------------
```

JCL de définition du fichier VSAM **EZACONFG** et configuration de la macro **EZACICD** pour l'environnement Sockets CICS

```
//*************************************************************//
  //*   THE FOLLOWING JOB DEFINES AND THEN LOADS THE VSAM     *//
  //*   FILE USED FOR CICS/TCP CONFIGURATION. THE JOBSTREAM   *//
  //*   CONSISTS OF THE FOLLOWING STEPS.                      *//
  //*    1). DELETE A CONFIGURATION FILE IF ONE EXISTS        *//
  //*    2). DEFINE THE CONFIGURATION FILE TO VSAM            *//
  //*    3). ASSEMBLE THE INITIALIZATION PROGRAM              *//
  //*    4). LINK THE INITIALIZATION PROGRAM                  *//
  //*    5). EXECUTE THE INITIALIZATION PROGRAM TO LOAD THE   *//
  //*        FILE                                             *//
  //*************************************************************//
  //PTCONFIG    JOB  MSGLEVEL=(1,1)
  //*
  //* THIS STEP DELETES AN OLD COPY OF THE FILE
  //* IF ONE IS THERE.
  //*
  //DEL    EXEC  PGM=IDCAMS
  //SYSPRINT DD  SYSOUT=*
  //SYSIN    DD  *
     DELETE -
       CICS.STM9.SOCKETS.CFG -
        PURGE -
        ERASE
  //*
  //*  THIS STEP DEFINES THE NEW FILE
  //*
  //DEFILE EXEC  PGM=IDCAMS
  //SYSPRINT DD SYSOUT=*
  //SYSIN    DD *
    DEFINE CLUSTER (NAME(CICS.STM9.SOCKETS.CFG) VOLUMES(CICSVOL) -
        CYL(1 1) -
        IMBED -
        RECORDSIZE(150 150) FREESPACE(0 15) -
        INDEXED ) -
        DATA ( -
          NAME(CICS.STM9.SOCKETS.CFG.DATA) -
          KEYS (16 0) ) -
        INDEX ( -
          NAME(CICS.STM9.SOCKETS.CFG.INDEX) )
   /*
  //*
//* THIS STEP ASSEMBLES THE INITIALIZATION PROGRAM
  //*
  //PRGDEF  EXEC PGM=IEV90,PARM='OBJECT,TERM',REGION=1024K
  //SYSLIB    DD DISP=SHR,DSNAME=SYS1.MACLIB
  //            DD DISP=SHR,DSNAME=TCPV32.SEZACMAC
  //SYSUT1    DD UNIT=SYSDA,SPACE=(CYL,(5,1))
  //SYSUT2    DD UNIT=SYSDA,SPACE=(CYL,(2,1))
  //SYSUT3    DD UNIT=SYSDA,SPACE=(CYL,(2,1))
  //SYSPUNCH  DD DISP=SHR,DSNAME=NULLFILE
  //SYSLIN    DD DSNAME=&&OBJSET,DISP=(MOD,PASS),UNIT=SYSDA,
  //             SPACE=(400,(500,50)),
  //             DCB=(RECFM=FB,BLKSIZE=400,LRECL=80)
  //SYSTERM   DD SYSOUT=*
  //SYSPRINT  DD SYSOUT=*
  //SYSIN     DD *
       EZACICD TYPE=INITIAL,         X
             PRGNAME=EZACICDF,       X
             FILNAME=EZACONFG
       EZACICD TYPE=CICS,            X
             TCPADDR=TCPIP,       X
             NTASKS=20,                      X
             DPRTY=10,                 X
             CACHMIN=10,               X
             CACHMAX=20,               X
             CACHRES=5,                X
             ERRORTD=CSKN,             X
```

```
            APPLID=A6ECCSM9
       EZACICD TYPE=LISTENER,  X
            TRANID=CSKL,              X
            PORT=9953,          X
  BACKLOG=40,           X
          ACCTIME=30,           X
          GIVTIME=10,           X
          REATIME=300,          X
          NUMSOCK=100,          X
          WLMGN1=CICSSTM9,   X
          MINMSGL=4,            X
          APPLID=A6ECCSM9
       EZACICD TYPE=FINAL
/*
 //*
 //* THIS STEP LINKS THE INITIALIZATION PROGRAM
 //*
 //LINK    EXEC PGM=IEWL,PARM='LIST,MAP,XREF',
 //             REGION=512K,COND=(4,LT)
 //SYSPRINT  DD SYSOUT=*
 //SYSUT1    DD SPACE=(CYL,(5,1)),DISP=(NEW,PASS),UNIT=SYSDA
 //SYSLMOD   DD DSNAME=&&LOADSET,DISP=(MOD,PASS),UNIT=SYSDA,
 //             SPACE=(TRK,(1,1,1)),
 //             DCB=(DSORG=PO,RECFM=U,BLKSIZE=32760)
 //SYSLIN    DD DSNAME=&&OBJSET,DISP=(OLD,DELETE)
 //*
 //* THIS STEP EXECUTES THE INITIALIZATION PROGRAM
 //*
 //FILELOAD EXEC PGM=*.LINK.SYSLMOD,COND=(4,LT)
 //EZACONFG DD  DSNAME= CICS.STM9.SOCKETS.CFG,DISP=OLD
```

Vous avez aussi la possibilité de configurer l'interface CICS Sockets TCP/IP par la transaction **EZAC**.

JCL de définition du fichier VSAM **EZACACHE** et configuration de la macro **EZACICR** pour utilisation du Cache DNS

```
//*******************************************************************//
  //*   THE FOLLOWING JOB DEFINES AND THEN LOADS THE VSAM             *//
  //*   FILE USED FOR THE CACHE.  THE DEFINITION CONSISTS OF          *//
  //*   TWO IDCAMS STEPS TO PERFORM THE VSAM DEFINITION               *//
  //*   AND A STEP USING EZACICR TO BUILD THE FILE LOAD               *//
  //*   PROGRAM. THE FINAL STEP EXECUTES THE FILE LOAD                *//
  //*   PROGRAM TO CREATE THE FILE.                                   *//

//*******************************************************************//
  //PTCACHE  JOB  MSGLEVEL=(1,1)
  //*
  //* THIS STEP DELETES AN OLD COPY OF THE FILE
  //* IF ONE IS THERE.
  //*
  //DEL    EXEC  PGM=IDCAMS
  //SYSPRINT DD  SYSOUT=*
  //SYSIN    DD  *
     DELETE -
      CICS.STM9.SOCKETS.EZACACHE -
        PURGE -
        ERASE
  //*
  //*  THIS STEP DEFINES THE NEW FILE
  //*
  //DEFILE EXEC  PGM=IDCAMS
  //SYSPRINT DD SYSOUT=*
  //SYSIN    DD *
    DEFINE CLUSTER (NAME(CICS.STM9.SOCKETS.EZACACHE) VOLUMES(CICSVOL) -
        CYL(1 1) -
        IMBED -
        RECORDSIZE(500 1000) FREESPACE(0 15) -
        INDEXED ) -
        DATA ( -
          NAME(CICS.STM9.SOCKETS.EZACACHE.DATA) -
          KEYS (255 0) ) -
        INDEX ( -
          NAME(CICS.STM9.SOCKETS.EZACACHE.INDEX) )
```

```
/*
  //*
  //*  THIS STEP DEFINES THE FILE LOAD PROGRAM
  //*
  //PRGDEF  EXEC PGM=IEV90,PARM='OBJECT,TERM',REGION=1024K
  //SYSLIB    DD DISP=SHR,DSNAME=SYS1.MACLIB
  //            DD DISP=SHR,DSNAME=TCPV32.SEZACMAC
  //SYSUT1    DD UNIT=SYSDA,SPACE=(CYL,(5,1))
  //SYSUT2    DD UNIT=SYSDA,SPACE=(CYL,(2,1))
  //SYSUT3    DD UNIT=SYSDA,SPACE=(CYL,(2,1))
  //SYSPUNCH  DD DISP=SHR,DSNAME=NULLFILE
  //SYSLIN    DD DSNAME=&&OBJSET,DISP=(MOD,PASS),UNIT=SYSDA,
  //             SPACE=(400,(500,50)),
  //             DCB=(RECFM=FB,BLKSIZE=400,LRECL=80)
  //SYSTERM   DD SYSOUT=*
  //SYSPRINT  DD SYSOUT=*


  //SYSIN     DD *
            EZACICR TYPE=INITIAL
            EZACICR TYPE=RECORD,NAME=ESSONVS1
            EZACICR TYPE=FINAL
/*
  //LINK    EXEC PGM=IEWL,PARM='LIST,MAP,XREF',
  //             REGION=512K,COND=(4,LT)
  //SYSPRINT  DD SYSOUT=*
  //SYSUT1    DD SPACE=(CYL,(5,1)),DISP=(NEW,PASS),UNIT=SYSDA
  //SYSLMOD   DD DSNAME=&&LOADSET(GO),DISP=(MOD,PASS),UNIT=SYSDA,
  //             SPACE=(TRK,(1,1,1)),
  //             DCB=(DSORG=PO,RECFM=U,BLKSIZE=32760)
  //SYSLIN    DD DSNAME=&&OBJSET,DISP=(OLD,DELETE)
  //*
  //*  THIS STEP EXECUTES THE FILE LOAD PROGRAM
  //*
  //LOAD EXEC PGM=*.LINK.SYSLMOD,COND=((4,LT,ASM),(4,LT,LINK))
  //EZACICRF DD DSN= CICS.STM9.SOCKETS.EZACACHE,DISP=OLD
```

Définition table PLT (*TCP/IP V320)

Pour démarrage/arrêt automatique de l'interface Sockets CICS (*TCP/IP V320) il est nécessaire d'ajouter le module **EZACIC20** dans la table **PLT**.

Pour le Démarrage automatique, ajoutez en **PLTPI** après l'entrée **DFHDELIM** :

```
DFHPLT TYPE=ENTRY,PROGRAM=EZACIC20
```

Pour l'arrêt automatique ajoutez en **PLTSD** avant l'entrée **DFHDELIM** :

```
DFHPLT TYPE=ENTRY,PROGRAM=EZACIC20
```

Configuration TCP/IP MVS/ESA

Modification configuration TCP/IP

Pour l'utilisation des Sockets CICS TCP/IP, il est nécessaire de définir un PORT pour la région CICS dans le fichier de configuration TCP/IP **hlq.PROFILE.TCPIP**.

```
;*********************************************
; PROFILE.TCPIP
; =============
;
; ------------------------------------------------------------------------
;
; NOTES:
;
;    A port that is not reserved in this list can be used by any user.
;    If you have TCP/IP hosts in your network that reserve ports
;    in the range 1-1023 for privileged applications, you should
;    reserve them here to prevent users from using them.
;
;    The port values below are from RFC 1060, "Assigned Numbers."
;
```

```
PORT
1415 TCP CSQ9CHIN            ; MQSeries CSQ9
9011 TCP PTMBRUNT            ; TeamConnection
9950 TCP EXCICS9             ; CICS Socket for CICS9
9953 TCP CICSSTM9            ; CICS Socket for A6ECCSM9
-------------------------------------------------------------------------
```

Paramètre TCPJOBNAME dans le fichier *hlq*.TCPIP.DATA

Pour l'initialisation de CICS TCP/IP, il est nécessaire de connaître le nom de la procédure TCP/IP MVS spécifié dans le fichier **hlq.TCPIP.DATA**, paramètre **TCPIPJOBNAME**.

```
;
;*********************************************************
;   Name of Data Set:     TCPIP.DATA                      *
;                                                         *
;   COPYRIGHT = NONE.                                     *
;                                                         *
; This data, TCPIP.DATA, is used to specify configuration *
;   information required by TCP/IP client programs.       *
;                                                         *
;                                                         *
;   Syntax Rules for the TCPIP.DATA configuration data set*
;       treated as a comment.                             *
;                                                         *
*********************************************************
; TCPIPJOBNAME specifies the name of the started procedure that was
; used to start the TCPIP address space.    TCPIP is the default.
;
TCPIPJOBNAME TCPIP
```

Démarrage et arrêt manuel du CICS TCP/IP

Démarrage du CICS TCP/IP

Exécutez la transaction CICS -> **CSKE** pour le démarrage manuel de CICS TCP/IP (*TCP/IP V310).

```
-------------------------------------------------------------------------
CSKE                                                          EZACIC00
                          CICS TASK-RELATED USER EXIT
                              CONNECTION MANAGER

                            ENABLE CICS-TCP/IP API


                         TCPIPJOBNAME tcpip--- PORT 9953-
 PF1=HELP          PF3=QUIT                                  EZAME00
-------------------------------------------------------------------------
```

Exécutez la transaction CICS -> **EZAO** pour le démarrage manuel de CICS TCP/IP (*TCP/IP V320).

```
-------------------------------------------------------------------------
EZAO,START,CICS

 APPLID=      ===> A6ECCSM9             APPLID of CICS

-------------------------------------------------------------------------
```

Ensuite, vérifiez par **CEMT I TAS** que la transaction **CSKL** est active :

```
CEMT I TAS
```

```
STATUS:  RESULTS - OVERTYPE TO MODIFY
Tas(0000023) Tra(CKAM)           Sus Tas Pri( 255 )
Tas(0000024) Tra(CKTI)           Sus Tas Pri( 001 )
Tas(0000027) Tra(DSNC)           Sus Tas Pri( 255 )
Tas(0000034) Tra(ISER)           Sus Tas Pri( 254 )
Tas(0000046) Tra(CSKL)           Sus Tas Pri( 255 )
```

Arrêt du CICS TCP/IP

Exécutez la transaction CICS => **CSKD** pour l'arrêt manuel de CICS TCP/IP (*TCP/IP V310).

```
-------------------------------------------------------------------------
CSKD                                                        EZACIC00
                                    CICS TASK-RELATED USER EXIT
                                    CONNECTION MANAGER

                                    DISABLE CICS-TCP/IP API


                                    ===> 1  QUIESCENT DISABLE-API
                                    ===> 2  IMMEDIATE DISABLE-API

              ENTER ONE OF THE ABOVE DISABLE-API OPTION NUMBERS: 2

PF1=HELP          PF3=QUIT                                  EZAMD00
-------------------------------------------------------------------------
```

Exécutez la transaction CICS => **EZAO** pour l'arrêt manuel de CICS TCP/IP (*TCP/IP V320).

```
-------------------------------------------------------------------------
EZAO,STOP,CICS

 APPLID    ===> A6ECCSM9            APPLID of CICS
 IMMEDIATE    ===>  Y                        Enter Yes!No
-------------------------------------------------------------------------
```

Compilation Cobol du programme Moniteur de Communication VA Pac

Modification du JCL de compilation et d'édition des liens pour l'intégration de l'interface CICS Socket TCP/IP.

```
//PTTDASOC  JOB (661),LH,CLASS=X,MSGCLASS=X,MSGLEVEL=(0,0)
//*JCLLIB ORDER=DSNY220.JCLLIB
//*************************************************
//*  Jcl de compil/link pour moniteur de communication VAP *
//*   avec l'interface CICS SOCKET TCP/IP                    *
//****************************************** **
//CBL EXEC DB2SOCK,MEMBER=CLTMVS,LOAD=PT$VIC.VIC.MTR8,
//          DBRMLIB=PT$VIC.VIC.DBRMLIB,SOURCE=PT$VIC.CICS.SOURCE
//LKED.SYSLIB  DD DSN=PT$VIC.TCPIP310.SEZATCP,LIB=SHR
//LKED.SYSIN DD *
    INCLUDE  SYSLIB(EZACICAL)
    INCLUDE  SYSLIB(EZACIC04)
    INCLUDE  SYSLIB(EZACIC05)
    INCLUDE  DB2LOAD(DSNCLI)
    NAME     CLTMVS(R)
 //
```

Définitions CICS pour l'application VA Pac

Définition du code Transaction

```
--------------------------------------------------------------------------------
TRansaction    : VSO1
   Group          : VISUAL
   DEscription    : VISUAL/CLIENT Transaction SOCKET CICS TCP/IP
   PROGram        : CLTMVS
   TWasize        : 00000
   PROFile        : DFHCICST
   PArtitionset   :
   STatus         : Enabled
   PRIMedsize     : 00000
   TASKDATALoc    : Below
   TASKDATAKey    : User
  REMOTE ATTRIBUTES
   DYnamic        : No
   REMOTESystem   :
   REMOTEName     :
   TRProf         :
+  Localq         :
--------------------------------------------------------------------------------
```

Définition du programme Moniteur de Communication

```
--------------------------------------------------------------------------------
PROGram        : CLTMVS
    Group          : VISUAL
    DEscription    : Moniteur de communication SOCKET CICS TCP/IP
    Language       : CObol
    RELoad         : No
    RESident       : No
    USAge          : Normal
    USElpacopy     : No
    Status         : Enabled
    RSl            : 00
    Cedf           : Yes
    DAtalocation   : Below
    EXECKey        : User
   REMOTE ATTRIBUTES
    REMOTESystem   :
 +  REMOTEName     :
+  Transid        :
    EXECUtionset   : Fullapi
--------------------------------------------------------------------------------
```

Définition du fichier de travail VSAM

```
--------------------------------------------------------------------------------
File           : FRVABI
    Group             : VISUAL
    DEScription    :
   VSAM PARAMETERS
    DSNAme         : PT$VIC.CICS.FRBIS
    Password          :
    Lsrpoolid          : 1
    DSNSharing     : Allreqs
    STRings            : 001
    Nsrgroup          :
   REMOTE ATTRIBUTES
    REMOTESystem   :
    REMOTEName     :
    RECORDSize     :
    Keylength         :
   INITIAL STATUS
 +  STAtus         : Enabled
--------------------------------------------------------------------------------
```

TUXEDO

Client

Tuxedo /WS version 6.x

La seule configuration à mettre en œuvre consiste à indiquer l'adresse du serveur et le port associé à l'application par l'intermédiaire de la variable d'environnement **WSNADDR**.

WSNADDR doit respecter la syntaxe suivante :

```
WSNADDR=0X0002ppppaaaaaaaa
        |   |   ↳ adresse IP sur huit chiffres en hexa
        |   ↳ N° de port sur quatre chiffres en hexa
        ↳ domaine AF_INET.
```

exemple : **WSNADDR=0X00020BB8C0060A5D**

```
        |   ↳ 192.6.10.93
        ↳ 3000
```

Si la version de Tuxedo /WS le permet (version 6.4 ou ultérieure), cette adresse peut être spécifiée sous la forme suivante :

```
WSNADDR=//serveur:port
        |   ↳ N° de port
        ↳Nom logique ou adresse IP du serveur
```

exemple : **WSNADDR=//9.143.96.178:3005**

Serveur

Se référer au manuel TUXEDO.

Le seul impératif à respecter est que les noms des services dans un serveur Tuxedo doivent correspondre aux noms externes des programmes.

MQSERIES

CICS Adapter

Exemple d'une configuration simple d'une application CICS déclenchée par le Trigger Monitor MQSeries.

- Définition **REQUEST QUEUE**

```
Queue name  . . . . . . . . . . VAP.REQUEST.Q
Description . . . . . . . . . : REQUEST QUEUE

Put enabled . . . . . . . . . : Y  Y=Yes,N=No
Get enabled . . . . . . . . . : Y  Y=Yes,N=No
Usage . . . . . . . . . . . . : N  N=Normal,X=XmitQ
Storage class . . . . . . . . : DEFAULT
Creation method . . . . . . . : PREDEFINED
Output use count  . . . . . . : 0
Input use count . . . . . . . : 0
Current queue depth . . . . . : 0
Default persistence . . . . . : Y  Y=Yes,N=No
Default priority  . . . . . . : 5  0 - 9
Message delivery sequence . . : F  P=Priority,F=FIFO
Permit shared access  . . . . : Y  Y=Yes,N=No
Default share option  . . . . : S  E=Exclusive,S=Shared
Index type  . . . . . . . . . : N  N=None,M=MsgId,C=CorrelId
Maximum queue depth . . . . . : 100000     0 - 999999999
Maximum message length  . . . : 4194304  0 - 4194304
Retention interval  . . . . . : 999999999  0 - 999999999 hours
Creation date . . . . . . . . : 1999-06-23
Creation time . . . . . . . . : 12.59.47

Trigger Definition

Trigger type  . . . . . . . . : E  F=First,E=Every,D=Depth,N=None

   Trigger set  . . . . . . . : Y  Y=Yes,N=No
   Trigger message priority : 0  0 - 9
   Trigger depth  . . . . . . : 1          1 - 999999999
   Trigger data . . . . . . . :

   Process name . . . . . . . : VAP.PROCESS.DEF
   Initiation queue . . . . . : CICS.INITQ
```

- Définition **REPLY QUEUE**

```
Queue name  . . . . . . . . . . VAP.REPLY.Q
Description . . . . . . . . . : output QUEUE


Put enabled . . . . . . . . . : Y  Y=Yes,N=No
Get enabled . . . . . . . . . : Y  Y=Yes,N=No
Usage . . . . . . . . . . . . : N  N=Normal,X=XmitQ
Storage class . . . . . . . . : DEFAULT
Creation method . . . . . . . : PREDEFINED
Output use count  . . . . . . : 0
Input use count . . . . . . . : 0
Current queue depth . . . . . : 0
Default persistence . . . . . : Y  Y=Yes,N=No
Default priority  . . . . . . : 0  0 - 9
Message delivery sequence . . : F  P=Priority,F=FIFO
Permit shared access  . . . . : Y  Y=Yes,N=No
Default share option  . . . . : S  E=Exclusive,S=Shared
Index type  . . . . . . . . . : N  N=None,M=MsgId,C=CorrelId
Maximum queue depth . . . . . : 999999999  0 - 999999999
Maximum message length  . . . : 4194304  0 - 4194304
Retention interval  . . . . . : 999999999  0 - 999999999 hours
```

```
Creation date . . . . . : 1999-06-23
Creation time . . . . . : 13.00.10

Trigger type  . . . . . : N  F=First,E=Every,D=Depth,N=None

   Trigger set  . . . . . : N  Y=Yes,N=No
   Trigger message priority : 0  0 - 9
   Trigger depth  . . . . . : 1          1 - 999999999
   Trigger data . . . . . :

   Process name . . . . . :
   Initiation queue . . . . :
```

- Définition **INITIATION QUEUE**

```
Queue name  . . . . . . . . . CICS.INITQ
Description . . . . . : CKTI initiation queue

Put enabled . . . . . : Y  Y=Yes,N=No
Get enabled . . . . . : Y  Y=Yes,N=No
Usage . . . . . : N  N=Normal,X=XmitQ
Storage class . . . . . : SYSTEM
Creation method . . . . . : PREDEFINED
Output use count  . . . . . : 0
Input use count . . . . . : 1
Current queue depth . . . . : 0
Default persistence . . . . : Y  Y=Yes,N=No
Default priority  . . . . . : 5  0 - 9
Message delivery sequence . : F  P=Priority,F=FIFO
Permit shared access  . . . : Y  Y=Yes,N=No
Default share option  . . . : E  E=Exclusive,S=Shared
Index type  . . . . . : N  N=None,M=MsgId,C=CorrelId
Maximum queue depth . . . . : 100        0 - 999999999
Maximum message length  . . : 4194304    0 - 4194304
Retention interval  . . . . : 999999999  0 - 999999999 hours
Creation date . . . . . : 1999-05-07
Creation time . . . . . : 18.30.18
Trigger Definition

Trigger type  . . . . . : N  F=First,E=Every,D=Depth,N=None

   Trigger set  . . . . . : N  Y=Yes,N=No
   Trigger message priority : 0  0 - 9
   Trigger depth  . . . . . : 1          1 - 999999999
   Trigger data . . . . . :

   Process name . . . . . :
   Initiation queue . . . . :
    Event Control
```

- Définition **PROCESS**

```
Process name . . . . . VAP.PROCESS.DEF
Description  . . . . . : VAP Process

Application type . . . . . : CICS
Application ID . . . . . : AMQM  ← Transaction application Serveur
User data  . . . . . : VAP.REQUEST.Q QMGR
```

MQSeries Client

Ce composant Client MQSeries doit être installé sur chaque poste de l'application Client et configuré via les variables d'environnement du système suivantes :

- **MQSERVER**

Cette variable d'environnement est un moyen simplifié pour indiquer l'**unique** chemin (MQI Channel) permettant d'atteindre le serveur.

Elle doit être définie sous les formes suivantes selon le protocole de transport utilisé :
. TCP/IP:
<ChannelName>/TCP/<ServeurName>(<PortNumber>)
Le numéro de port par défaut est 1414

. LU 6.2:
<ChannelName>/LU62/<SymbolicDestName>/<ModeName>/<TPName>

Exemple sous Unix:
```
set MQSERVER = CHANNEL.TO.VAP/tcp/9.134.12.87(1414)
export MQSERVER
```

- **MQCHLLIB** et **MQCHLTAB**

  Lorsque, sur une station de travail, les applications MQSeries doivent utiliser plus d'un Channel, il n'est plus possible d'utiliser la variable d'environnement **MQSERVER**. L'administrateur du réseau MQSeries doit générer et fournir un fichier de description des Channels (**amqclchl.tab** par défaut) et le rendre accessible pour les postes Client.

  Les variables d'environnement **MQCHLLIB** et **MQCHLTAB** permettent d'indiquer respectivement le chemin de localisation et le nom de ce fichier.

TDS-TCP/IP

Installation / configuration TDS-TCP/IP client

Installation

Sur **Windows NT**, vous devez installer:

- le fichier **Atmi32.dll** dans un répertoire accessible par le PATH (ex: répertoire du middleware VAP).
- le fichier **Atmi.ini** dans le répertoire système de Windows (ex: c:\winnt).

Sur **AIX 4.3**, vous devez :

- Transférer le fichier **XATMI** vers AIX,
- Spécifier le nom du répertoire, où **XATMI** est placé, dans la variable d'environnement **LIBPATH**

  Ex: **export LIBPATH=$LIBPATH:/vap/middleware**

Fichiers host et services

Dans le répertoire **c:\winnt\system32\drivers** pour Windows NT et **/etc** pour Unix.

Dans le fichier **hosts**, vous devez définir une entrée pour le serveur GCOS7 (ex : **213.62.98.213 bc0e** #GCOS7)

Dans le fichier **services**, vous devez définir le service dont le nom résulte de la concaténation du nom de **host** avec le nom du TDS (ex : **bc0evpd5 51000/tcp** #TCP/IP Access TDS)

Traces ATMI

Fichier ATMI.INI pour Windows

Ce fichier doit être placé dans le répertoire système de Windows (ex: c:\winnt), et contient les paramètres suivants :

```
[ATMI]

; PATH : pour les fichiers de log et de trace (chemin courant par défaut)

; CRYPT : obligatoire, ne peut pas être modifié
CRYPT=YES

; DEBUG : echo des messages de warning
  0 : pas d'affichage des messages
  1 : affichage de tous les messages, sauf les messages de déconnexion
  2 : affichage de tous les messages
DEBUG=0

; TRACE_API : trace utilisateur: paramètres d'entrée et sortie des
  paramètres des fonction XATMI appelées
```

0 : pas de trace
1 : trace des paramètres mais pas contenu du buffer
2 : trace des paramètres et contenu du buffer

```
TRACE_API=2
```

; **TRACE_SOC** : trace interne pour l'interface de debugging avec Windows Sockets
0 : pas de trace
1 : trace des fonctions appelées et des paramètres d'entrée et sortie
2 : 1 + contenu de l'en-tête échangé avec TDS

```
TRACE_SOC=0
```

; **TIMEOUT** : durée de la connexion en millisecondes
0 : pas de timeout

```
TIMEOUT=30000
```

Variables d'environnement pour AIX

Le fichier trace est : **ATMITDS.TRC_<process id>.**

Vous devez positionner les variables d'environnement suivantes :

ATMI_TRACE_API : trace des appels API.
0 : (défaut) pas de trace.
1 : trace des premiers 32 octets des buffers.
2 : trace de tout le contenu des buffers.

☞ La valeur **0** est généralement recommandée. La valeur **1** est recommandée en cas de problème pour lequel le contenu des buffers d'émission / réception n'est pas nécessaire. La valeur **2** doit être réservée en cas de problème pour lequel le contenu des buffers d'émission / reception doit être examiné.

ATMI_TRACE_SOC : trace des verbes socket. Cette variable est réservée au debugging système.
0 : (défaut) pas de trace.
1 : trace des verbes socket.
2 : trace des en-têtes d'émission / reception échangés avec TDS.

↳ Ces variables d'environnement doivent être exportées de façon standard pour que l'application puisse accéder à leurs valeurs.

De plus, indépendamment du mécanisme de fichier trace, vous pouvez demander la log des messages d'erreur via la variable d'environnement **ATMI_DEBUG**.
0 : (défaut) pas d'affichage console.
1 : seuls les messages de déconnexion anormale sont affichés sur la console.
2 : affichage de toutes les erreurs sur la console.

Exemple de mise en œuvre sous TDS

Exemple de source TDS

Nom du TDS : **VPD5**

Source TDS : membre **STDS** de la bibliothèque **VPD5.SLLIB**

L'utilisation du Socket TCP/IP doit être déclarée dans le membre **STDS** sous **VPD5.SLLIB** et une nouvelle génération TDS (**TP7GEN**) doit être effectuée.

Vous pouvez modifier le membre **STDS** de **VPD5.SLLIB** comme suit :

```
 TDS SECTION.
 PROGRAM-ID. VPD5.
* BTNS                IS BTNS.
 NUMBER TERMINALS       15.
 NUMBER OF DUMMY CORRESPONDENT IS 1 MAXIMUM IS 3.
 SIMULTANEITY           10.
 RESERVE                280 AREAS.
 ATTACH SHARABLE MODULE H_SM_DCM.
 NUMBER MODULES         10.
 MESSAGE-LENGTH         32001.
 TPR-TIME-LIMIT         500000.
 TCP-IP PROTOCOL USED WITH OPEN7.
 USE "MENU"     TRANSACTION-MENU.
 USE ZAR990.
 USE ZARS12.
 USE FORMS.
 USE CLCLNT.
 USE CLPROD.
 SERVICE-MESSAGE HEADER IS "27F1C3"
     TRAILER IS "4040".
 CANCELCTX AT RECONNECTION.
 ...
```

Vous définissez ensuite dans **STDS** la transaction qui lance le Moniteur de Communication :

```
 TRANSACTION SECTION.
 MESSAGE "VTCP" ASSIGN    CLSOCK
     IMPLICIT COMMITMENT
     PAGES                 50
     WITH TPR ACCOUNTING
     AUTHORITY-CODES       31
     TRANSACTION-STORAGE   SIZE  500.
 ...
```

Référence : DDOVM000352F